TECHNISCHE
UNIVERSITÄT
DARMSTADT

INFORMATIONSBESCHAFFUNG AUS DIGITALEN TEXTRESSOURCEN —

Domänenadaptive Verfahren zur Strukturierung heterogener Textdokumente

Vom Fachbereich Elektrotechnik und Informationstechnik
der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte Dissertation

von

SEBASTIAN SCHMIDT, M.SC.

Geboren am 30. August 1985 in Gießen

Vorsitz: Prof. Dr.-Ing. Marius Pesavento
Referent: Prof. Dr.-Ing. Ralf Steinmetz
Korreferent: Prof. Dr.-Ing. Michael Zink

Tag der Einreichung: 03. November 2015
Tag der Disputation: 21. Dezember 2015

Hochschulkennziffer D17
Darmstadt 2016

Getting information off the Internet
is like taking a drink from a fire hydrant.

— Mitchell Kapor

KURZFASSUNG

In der heutigen Informationsgesellschaft sind Personen häufig mit der sogenannten *Informationsüberflutung* konfrontiert. Dies bedeutet, dass es aufgrund der enormen Menge insbesondere digital verfügbarer textueller Ressourcen zu einer Überforderung bei der Identifikation relevanter Informationen kommen kann. Bislang ist eine Unterstützung bei dieser Aufgabe vorrangig über Volltextsuchen in Textsammlungen möglich, die jedoch keine komplexen Suchanfragen mit Beschreibung unterschiedlicher Aspekte der Suchanfrage erlauben. Werkzeuge zur elaborierten Suche, welche es erlauben, einzelne Aspekte der zu suchenden Information zu beschreiben, existieren nur in spezifischen Domänen. Ein wesentlicher Grund hierfür ist, dass die zu durchsuchenden digitalen Textressourcen meist in unstrukturierter Form vorliegen. Damit ist kein einheitlicher, gezielter Zugriff auf spezifische Informationen innerhalb der Dokumente möglich, welcher die Realisierung solcher Werkzeuge vereinfachen würde. Strukturierte Repräsentationen der Dokumente, in denen die Bedeutung einzelner Textfragmente für die in den Dokumenten beschriebenen Entitäten zu erkennen ist, würden diesen Zugriff ermöglichen.

Im Rahmen dieser Dissertation wird untersucht, mit welchen Verfahren textuelle Dokumente automatisiert in eine strukturierte Repräsentation überführt werden können. Existierende Ansätze mit gleicher oder ähnlicher Zielsetzung sind meist für spezifische Anwendungsdomänen entwickelt und lassen sich nur schwer in andere Domänen übertragen. Bei Einsatz in neuen Domänen müssen bislang somit vollständig neue Ansätze zur Strukturierung entworfen werden oder zur Übertragung von Ansätzen ein großer manueller Aufwand erbracht werden. Daraus resultiert die Notwendigkeit, domänenadaptive Verfahren zur Strukturierung von Textressourcen zu entwickeln. Dem steht als wesentliche Herausforderung die Heterogenität von Anwendungsdomänen hinsichtlich verschiedener Kriterien wie verwendeter Dokumentenformate, vorherrschender Textlänge und domänenspezifischer Terminologie entgegen.

Die Untersuchung von fünf ausgewählten heterogenen Anwendungsdomänen zeigte, dass bestimmte Typen von Informationen domänenübergreifend von Relevanz sind. Daher wurden für drei dieser Typen Verfahren konzipiert, welche Informationen dieser Typen in heterogenen Dokumenten identifizieren können. Hierbei wurde sichergestellt, dass für die erstmalige Anwendung der Verfahren in einer spezifischen Domäne möglichst wenig manueller Aufwand erforderlich ist, um die Anforderung der Domänenadaptivität der Verfahren zu berücksichtigen. Zur Reduktion des manuellen Aufwands wurden Techniken des maschinellen Lernens, wie der Ansatz des *Active Learning*, sowie existierende, frei verfügbare Wissensbasen verwendet. Die konzipierten Verfahren wurden implementiert und unter Verwendung von Textkorpora aus den zuvor analysierten Domänen evaluiert. Dabei konnte gezeigt werden, dass die Identifikation von Informationen dieser drei Typen mit hoher Güte möglich ist und gleichzeitig eine gute Domänenadaptivität erreicht wird. Weiterhin wurden unabhängige Verfahren zur Identifikation von Informationen der einzelnen Typen kombiniert, um eine Strukturierung kompletter Dokumente durchführen zu können. Dieses Konzept wurde in einer Fallstudie für eine der Anwendungsdomänen imple-

mentiert und unter Verwendung eines Textkorpus aus dieser Domäne evaluiert. Die Resultate bestätigen, dass eine Strukturierung mittels Kombination der Verfahren zur Identifikation der Informationen der einzelnen Typen erreicht werden kann.

Unter Verwendung der in dieser Dissertation vorgestellten domänenadaptiven Verfahren lassen sich strukturierte Repräsentationen aus unstrukturierten digitalen Textressourcen erstellen, die die vereinfachte Realisierung von Werkzeugen zur Informationsbeschaffung ermöglichen. Die daraus resultierenden Möglichkeiten für elaborierte Werkzeuge zur Informationsbeschaffung reduzieren die Überforderung der Nutzer bei der Identifikation relevanter Informationen.

ABSTRACT

In today's information society, users are increasingly confronted with the so-called "*information overload*" problem. They are often overwhelmed by the huge amount of mostly digitally available textual resources when trying to identify relevant information suiting their information needs.

So far, users are mainly left only with a full-text search due to the lack of more elaborate tools which would allow them to specify different aspects of their information need. Elaborate search tools, that allow a precise definition of information needs, only exist in specific domains. One of the main reasons is that the mostly unstructured nature of digital textual resources does not allow access to specific information within the documents which would enable the realization of these tools. A structured representation of the documents, where the meaning of individual text fragments for the entities being described in the documents is known, would allow for this access.

The goal of this thesis is to investigate approaches that would automatically transform documents into structured representations. Existing approaches that have similar aims are often tailored to specific applications and thus cannot be easily applied to other applications or domains. Their deployment in new domains currently requires a redesign of the approaches or significant manual effort for their adaptation. Based on this observation, this thesis aims to develop domain-adaptive approaches to structure textual documents. A major challenge for the design of appropriate methods is the heterogeneity of application domains, in particular with regards to the document formats, lengths of texts, and domain-specific terminology used.

A study of five selected heterogeneous domains revealed the existence of common types of information across domains. As a result of this finding, different methods were designed to identify information in heterogeneous documents for three of these types. As a design requirement, it was considered that only little manual effort is accepted when deploying the methods to a new domain. This requirement enables a good domain adaptation of the methods. In order to reduce the manual effort needed, techniques from the field of machine learning, such as *Active Learning*, were applied. Furthermore, freely available and domain-independent knowledge bases were integrated. The approaches were implemented and evaluated using data sets from the observed domains. Results showed that the identification of information of individual types is possible while still maintaining a good domain adaptivity. Finally, a concept was presented that combines methods for the identification of information with the goal of structuring entire documents. An implementation and evaluation of this concept revealed that structuring can be obtained through a combination of different methods, whereby each method identifies only a single type of information.

The domain adaptive means presented in this dissertation enable the creation of structured representations from unstructured digital textual resources. This simplifies the realization of various tools for information retrieval. The resulting possibilites for the development of new information retrieval tools reduce the overload problem experienced by users when trying to identify relevant information.

DANKSAGUNG

Diese Arbeit wäre kaum möglich gewesen ohne die riesige Unterstützung, praktischer und mentaler Art, die ich in den letzten vier Jahren durch verschiedene Leute bekommen habe.

Zunächst möchte ich meinen herzlichen Dank an Prof. Ralf Steinmetz für die Betreuung meiner Dissertation und die hervorragende Arbeitsumgebung bei KOM aussprechen. Mein Gruppenleiter Dr. Christoph Rensing hat mir insbesondere durch regelmäßige Treffen immer wieder geholfen auf den rechten Pfad zu kommen, er hat sich intensiv mit meinen Arbeiten auseinandergesetzt und mir viele wertvolle Kommentare geliefert. Vielen Dank, Christoph! Ein herzliches Dank geht auch an meinen Zweitgutachter Prof. Michael Zink, von dem ich wertvolles Feedback zu meiner Arbeit erhalten habe. Mein Mentor Dr. Andreas Faatz hat meinen Dissertationsprozess mit großem Interesse begleitet und mir regelmäßig seine sehr hilfreiche „Sicht von außen“ gegeben.

Weiterhin möchte ich meinen ehemaligen und aktuellen KOM-Kollegen und insbesondere meinen Kollegen in der Arbeitsgruppe *Knowledge and Educational Technologies* für ihre Unterstützung danken. Philipp hat mich als Betreuer meiner Masterarbeit für die Themen in der Arbeitsgruppe begeistern können, mit Moji und Renato habe ich viele sehr wertvolle und inspirierende Gespräche geführt und gleichzeitig viel Spaß gehabt beim Diskutieren nigerianischer, kolumbianischer und deutscher Eigenarten. Doreen hat immer ein offenes Ohr für mich gehabt und gemeinsame Zigaretten-Obst-Pausen mit Stephan in der Sonne haben Ablenkung und Inspiration gebracht. Ich bin sehr dankbar, dass ich Steffen als Studenten für eine Masterarbeit und als anschließenden Kollegen gewinnen konnte. Die Zusammenarbeit hat meine Dissertation wesentlich voran getrieben. Außerdem gab es mit ihm und Irina im Büro immer viel zu lachen. Lena hat für Süßigkeitennachschub gesorgt, Johannes und Andreas R. haben mir in der finalen Phase wertvolles Feedback zu meiner Arbeit gegeben. Vielen Dank euch allen!

Vielen Dank an Sebastian Wollny sowie meine anderen Studenten und HiWis für ihren Input. Außerdem ein herzliches Dankeschön an die kimeta GmbH, insbesondere Sven Kloppenburg und Elena Neuschild, für die hervorragende Zusammenarbeit über die letzten Jahre.

Aber auch jenseits des Arbeitsumfeldes habe ich viel Unterstützung für meine Arbeit bekommen. An dieser Stelle möchte ich insbesondere meinen Freunden einen großen Dank aussprechen für Ablenkungen unterschiedlichster Art, Verständnis, wenn mich meine Arbeit zu sehr eingenommen hat, und Interesse an meiner Dissertation. Danke insbesondere an Flo K., Joachim, Caro, Franzi, Christian, Marie, Timm, Betim, Seppel, Raschid und Flo S.!

Ein besonders großer Dank geht an meine Familie. Meine Eltern unterstützen mich unendlich viel in jedem Lebensabschnitt und haben in den vergangenen vier Jahren viel Verständnis dafür gezeigt, wenn mal wieder keine Zeit war, meine Schwester Steffi, ihr Mann Ralf und insbesondere auch ihre Kinder Lara und Mika haben

für dankbare Ablenkung in Darmstadt gesorgt und mein Onkel Thomas hat immer großes Interesse an dem, was ich tue, und mich mit seinen Visionen inspiriert.

Mein größter Dank geht an Claudia. Danke, dass du das Abenteuer Dissertation zusammen mit mir durchlebt hast, mir dabei geholfen hast, wo du nur konntest, und dass du immer für mich da bist!

Darmstadt, 2016 Sebastian Schmidt

INHALTSVERZEICHNIS

1

EINFÜHRUNG

1.1 MOTIVATION

Sowohl im beruflichen Kontext als auch im privaten Umfeld ist ein regelmäßiges Zugreifen auf textuelle Medien zur Informationserlangung alltäglich geworden. Offensichtlich ist dies beispielsweise bei Wissensarbeitern, die sich mit der Lösung von individuellen, nicht-standardisierten Fragestellungen beschäftigen (vergleiche [132]) und daher nicht auf reines Erfahrungswissen zurückgreifen können. Aber auch in anderen Situationen muss auf textuelle Ressourcen, wie beispielsweise Berichte, Dokumentationen, Bedienungsanweisungen oder Nachschlagewerke zurückgegriffen werden, um Unterstützung bei der Beantwortung von Fragestellungen zu erhalten.

Aufgrund der ubiquitären Nutzung und Bereitstellung von Informationen mittels Technologie ist die Menge der für einen Einzelnen potentiell direkt verfügbaren textuellen Informationen stark gestiegen. Insgesamt wächst die Zahl verfügbarer Informationen in digitaler Form stetig mit steigender Geschwindigkeit [56, 68].

Der Informationsbedarf, welcher den Zugriff auf Sammlungen textueller Ressourcen auslöst, kann häufig nicht unmittelbar befriedigt werden. Einen wesentlichen zeitlichen Anteil bei der Informationsbeschaffung nimmt der Prozess der Suche nach Relevantem in Anspruch. Der Informationsbedürftige ist zunehmend überfordert, relevante Informationen aufzufinden und geeignet zueinander in Beziehung zu setzen. Zur Umschreibung dieser Problematik wurde der Begriff des *Information Overload* („Informationsüberflutung“) [11] geprägt.

Diese Schwierigkeit bei der Informationsbeschaffung sorgt zum einen für Unzufriedenheit beim Suchenden und zum anderen zu ökonomischen Einbußen aufgrund des hohen Zeitaufwands bei der Suche.

Drei wesentliche Faktoren sind für die Schwierigkeit bei der Informationsbeschaffung verantwortlich:

1. Die enorme Menge an textuellen Dokumenten ist einerseits vorteilhaft, da hierin sehr große Informationspotenziale stecken, andererseits erschwert diese Menge das Auffinden relevanter Textfragmente zur Erfüllung des Informationsbedarfs.

2. Es existiert keine globale und einheitliche Struktur zwischen einzelnen Dokumenten, die den Bezug zwischen ihnen widerspiegelt und dem Suchenden somit Unterstützung bei der Suche nach Referenzen bietet.

3. Ein Großteil der textuellen Dokumente besitzt keine einheitliche inherente Struktur, welche das Auffinden der benötigten Informationen innerhalb der einzelnen Dokumente vereinfachen würde.

Klassische *generische Suchmaschinen*, wie beispielsweise Google[1] oder bing[2] adressieren die beiden erstgenannten Faktoren teilweise. Als Grundlage dient eine Indizierung der im Internet verfügbaren Webseiten auf Basis des darin enthaltenen textuellen Inhalts und vergebener Metadaten wie Schlüsselwörter und Seitentitel. Weiterhin spielt die Verlinkungsstruktur zwischen Webseiten eine wesentliche Rolle bei der Bestimmung der Relevanz der einzelnen Suchergebnisse und somit der Ordnung der Suchergebnisse [121]. Inhaltliche Interpretationen und Zusammenhänge zwischen Informationen bleiben hierbei außen vor.

In der Vergangenheit hat sich das Konzept der *domänenspezifischen Suchmaschinen* [175] herausgebildet. Hierbei wurden die Defizite der generischen Suchmaschinen als Marktpotenzial identifiziert und Suchmaschinen entwickelt, die durch Beschränkung auf eine einzelne Domäne und die Nutzung von Wissen über diese Domäne einen Mehrwert gegenüber generischen Suchmaschinen bieten können. Mechanismen wie die *facettierte Suche* [167] erlauben es dem Nutzer, seine Suchanfrage präziser und somit zielorientierter zu stellen, als es mit einer, bei generischen Suchmaschinen üblichen, reinen Volltext-Suche möglich wäre. Zur Ermöglichung der facettierten Suche wird auf anwendungsspezifische Typen von Informationen, wie beispielsweise Produktnamen, Adressdaten oder behandelte Themen, die in den textuellen Dokumenten wörtlich zu finden sind oder abgeleitet werden können, zugegriffen. Um dies erreichen zu können, ist es jedoch notwendig, dass das benötigte Wissen über die entsprechende Domäne maschinennutzbar erstellt und aktuell gehalten wird, was mit einem hohen manuellen Aufwand verbunden ist. Ohne dieses Wissen ist die Klassifikation der für mögliche Suchanfragen relevanten Informationen in Dokumenten nicht mit hoher Güte umsetzbar. Weiterhin muss das Wissen über die beschriebenen, in der Anwendung relevanten Typen von Informationen vorhanden sein, um diese in der Suchlogik nutzen zu können.

Um den Aufwand bei der Filterung von Informationen gering zu halten, wurden daher neben den manuellen Verfahren (teil-)automatisierte Verfahren entwickelt, welche eine Klassifikation von ganzen Texten oder textuellen Aspekten vornehmen. Bisherige automatisierte Ansätze zur Klassifikation von textuellen Informationen, welche zur Klasse der überwachten Lernverfahren gehören, benötigen zur Erzielung einer hohen Genauigkeit große Mengen annotierter Beispieldaten [1], welche Domänenwissen in statistischer Form repräsentieren. Wenn diese Verfahren von einer Anwendungsdomäne in eine andere Anwendungsdomäne übertragen werden sollen, muss dieser initiale Aufwand der Auszeichnung von Beispieldaten erneut aufgebracht werden. Aufgrund dieses Aufwandes eignen sich diese Verfahren weniger gut zur automatisierten Identifikation relevanter Informationen in wechselnden Anwendungsdomänen.

Von Tim Berners-Lee wurde 2001 die Vision des *Semantic Web* begründet [14], welche Webseiten nicht nur auf struktueller Ebene mittels Hyperlinks, sondern auch auf semantischer Ebene miteinander verbindet. Ein wesentliche Voraussetzung dafür ist die fein-granulare semantische Annotation von Webseiten beziehungsweise der darin enthaltenen textuellen Informationen. Eine konsequente Nutzung von umfassenden Annotationen für Webseiten könnte das Auffinden relevanter Informationen wesentlich vereinfachen, da die Annotationen zur Identifikation relevanter In-

1 http://www.google.de, letzter Zugriff am 19.09.2015
2 http://www.bing.com, letzter Zugriff am 19.09.2015

formationen und zur Vernetzung zwischen Dokumenten genutzt werden könnten. Die Suche könnte beispielsweise auf Textfragmente eines spezifischen semantischen Typs reduziert werden. Der Prozess der manuellen Annotation ist aufgrund des hohen Aufwands für einzelne Webseiten und der großen Anzahl der Webseiten jedoch nicht praktisch realisierbar [139]. Diese Hürde bei der manuellen Annotation bremst die praktische Umsetzung des Gedankens des Semantic Web. So enthielten im Jahr 2013 bei einer Untersuchung von 12 Milionen Webseiten nur circa 26% der Webseiten semantische Annotationen, wobei häufig nur wenige Elemente auf den einzelnen Seiten annotiert waren und nur 18 semantische Annotationen konsistent auf mindestens 70 verschiedenen Webseiten verwendet wurden [107]. Dadurch kann de facto nicht auf semantische Annotationen als Strukturelemente zurückgegriffen werden und die relevanten Informationen müssen zu einem großen Teil aus unstrukturierten Dokumenten extrahiert werden.

Nicht nur zur Unterstützung der effizienten domänenspezifischen Suche im Internet ist Domänenwissen notwendig. Dies trifft ebenso auf die Suche in lokalen oder unternehmensinternen textuellen Datenquellen zu. Auch die darin enthaltenen Dokumente liegen häufig nur in unstrukturierter Form vor. Somit stellen sich auch hier die oben beschriebenen Herausforderungen bei der Suche nach Informationen in diesen Dokumentensammlungen.

1.2 ZIEL, ANSATZ UND BEITRÄGE DER DISSERTATION

Das Ziel der vorliegenden Dissertation ist die Konzeption und Evaluation von Verfahren zur Strukturierung textueller Dokumente. Als strukturierte Form eines Dokumentes wird in dieser Dissertation eine Repräsentation eines Dokumentes verstanden, in der die semantische Rolle, also die Bedeutung, einzelner Textfragmente zur Charakterisierung der in diesem Dokument beschriebenen Entität angegeben ist. Eine solche Repräsentation ermöglicht ein besseres Auffinden konkreter Informationen, da durch Anwendungen gezielt auf einzelne Informationen in Textfragmenten, deren semantische Rolle bekannt ist, im Dokument zugegriffen werden kann und nicht im kompletten Dokument gesucht werden muss.

Der Prozess der Strukturierung ist gemäß des Modells von Steinmetz [161] zur Klassifikation von multimedialen Anwendungen, als *Medienbearbeitung* zu bezeichnen. Somit steht dieser Prozess zwischen der initialen *Medienaufbereitung*, in der die Medien, wie beispielsweise textuelle Dokumente, initial erstellt werden, und der *Medienintegration*, in der die bearbeiteten Medien miteinander verknüpft werden.

Die Strukturierung erfordert die Identifikation relevanter Informationen in textuellen Dokumenten sowie die Zusammenführung dieser Informationen in eine gemeinsame Repräsentation. Die zu entwickelnden Verfahren sollen sich durch geringe Anforderungen an den manuellen Aufwand bei der Adaption an eine neue Domäne bei gleichzeitig hoher Genauigkeit bei der Identifikation von Informationen auszeichnen. Betrachtet werden sollen hierbei Mengen von Dokumenten, die jeweils aus einer gemeinsamen Anwendungsdomäne stammen und für Anwendungen in dieser Domäne relevant sind. Einerseits sollen also Dokumente, die aus einer Domäne stammen, mit einer hohen Genauigkeit anhand der für die betrachtete Domäne relevanten Struktur strukturiert werden. Andererseits sollen die zu entwickelnden Verfahren jedoch in ihrer Anwendung nicht auf einzelne Domänen beschränkt sein, sondern gut

von einer Domäne in eine andere Domäne adaptierbar sein. Eine wesentliche Herausforderung bei der Strukturierung und der Adaptierbarkeit ist die Heterogenität der Dokumente hinsichtlich Format, Länge und Inhalt, sowohl zwischen verschiedenen Domänen als auch innerhalb einer spezifischen Domäne. Diese Heterogenität resultiert aus den heterogenen Quellen der einzelnen Dokumente. So wurden die Dokumente beispielsweise von verschiedenen Erstellern auf verschiedenen Webseiten veröffentlicht. Diese berücksichtigen in der Regel auch keine vorgeschlagenen Standardprozesse zur Erstellung eines idealen Layouts für die Präsentation von Informationen [10, 131], welches eine gezielte Extraktion dieser Informationen vereinfachen würde.

Im Gegensatz zur zuvor beschriebenen Zielsetzung des Semantic Web, in der eine feingranulare Annotation von Webseiten sowie eine Verlinkung der annotierten Elemente erforderlich ist, werden im Rahmen dieser Arbeit Methoden entwickelt, um Dokumente sinnvoll zu strukturieren, so dass nur relevante Informationseinheiten für eine konkrete Anwendung zu Rate gezogen werden müssen. Eine Interpretation der Informationseinheiten mittels derer Verlinkung liegt nicht im Fokus. Die Beiträge der Arbeit liegen in diesem Strukturierungsschritt und nicht auf der Nutzung der strukturierten Daten. Die erzeugte strukturierte Datenrepräsentation kann für unterschiedliche Anwendungen, wie beispielsweise Suchmaschinen, Empfehlungssysteme oder Informationsmanagementsysteme genutzt werden.

Im Folgenden sind die zentralen Beiträge der vorliegenden Dissertation beschrieben:

- Zunächst wird eine Literaturanalyse vorgenommen, in der Arbeiten mit ähnlichen Zielsetzungen wie die vorliegende Dissertation klassifiziert und auf ihre Verwendbarkeit für die Ziele dieser Dissertation untersucht werden.

- Um Abhängigkeiten zwischen Domänen, Dokumentensammlungen, Dokumenten und den darin enthaltenen Informationen darstellen zu können, wird ein domänenunabhängiges Modell vorgestellt, welches diese Abhängigkeiten beschreibt.

- Es wird eine Analyse von potentiellen Anwendungsdomänen vorgenommen um deren Charakteristika bei der Entwicklung der Verfahren zur Dokumentenstrukturierung zu berücksichtigen. Für die Untersuchungen werden fünf heterogene Anwendungsdomänen ausgewählt, deren Auswahl mit dem Ziel der Orthogonalität der jeweiligen Eigenschaften stattfindet. So werden beispielsweise Domänen mit Dokumenten aus dem Internet einerseits oder aus Unternehmen andererseits ausgewählt. Weiterhin werden sowohl Domänen, in denen Textdokumente mit einer gewissen Vorstrukturierung im Vordergrund stehen, als auch Domänen, in denen völlig unstrukturierte Dokumente von Relevanz sind, betrachtet. Die betrachteten Domänen sind Stellenanzeigen, Impressumsseiten, Ausschreibungen studentischer Abschlussarbeiten, Kurzfassungen wissenschaftlicher Publikationen und Scrum-Protokolle.

- Basierend auf der Auswahl der Domänen werden insgesamt fünf Typen von relevanten Informationen identifiziert und charakterisiert. Diese charakterisierten Attributtypen sind Eigennamen, numerische Attribute, Freitextattribute, aggregierte Attribute und Meta-Attribute. Diese Informationen sind sowohl

wörtliche, aus den Dokumenten stammende, Textabschnitte unterschiedlicher Länge als auch Informationen, die aus dem textuellen Inhalt der Dokumente abgeleitet werden können. Weiterhin wird analysiert, welche Anforderungen die möglichen Methoden zur automatisierten Identifikation dieser Informationen erfüllen müssen.

- Ausgehend von den definierten Anforderungen werden Methoden zur Identifikation von drei Attributtypen konzipiert, die die in der Literaturanalyse identifizierten Lücken in existierenden wissenschaftlichen Arbeiten adressieren. Die Konzeption beachtet dabei die beiden genannten Ziele der Minimierung des notwendigen manuellen Aufwands zur Domänenadaption bei Maximierung der Genauigkeit der Methoden bei Anwendung in den einzelnen Domänen. Die Methoden werden anschließend als Softwarekomponenten implementiert und unter Verwendung von Evaluationskorpora aus den Anwendungsdomänen evaluiert. Die Methoden sind in unterschiedlichen Sprachen anwendbar, im Rahmen dieser Dissertation werden deutsch- und englischsprachige Dokumente betrachtet.

- Da sich im Regelfall mehrere relevante Informationen in einem einzelnen Dokument befinden, wird eine Kombination der Verfahren zur Identifikation der einzelnen Informationen benötigt. Ein solches Konzept zur Kombination wird entwickelt, prototypisch umgesetzt und in einer Anwendungsdomäne evaluiert.

1.3 GLIEDERUNG DER DISSERTATION

Im Folgenden wird die Gliederung der vorliegenden Dissertation beschrieben. Nach diesem Kapitel wird auf Grundlagen, die für das Verständnis der vorliegenden Arbeit notwendig sind, eingegangen (Kapitel 2). Weiterhin werden die Ergebnisse der Literaturanalyse vorgestellt. Im Rahmen dieser werden verwandte Arbeiten, die sich mit ähnlichen Zielsetzungen wie diese Dissertation beschäftigen, klassifiziert und eine Abgrenzung zu diesen vorgenommen (Kapitel 3). In Kapitel 4 wird das im Rahmen dieser Arbeit entwickelte Modell vorgestellt auf dessen Basis die konkreten Verfahren umgesetzt werden. Außerdem werden in diesem Kapitel die einzelnen betrachteten Anwendungsdomänen beschrieben und analysiert sowie die relevanten Typen von Informationen, die Attributtypen, charakterisiert.

In den folgenden Kapiteln werden Verfahren vorgestellt, um die adressierten Typen von Informationen in Texten zu identifizieren, weiterhin werden die Evaluationen der Verfahren und deren Ergebnisse in den Anwendungsdomänen beschrieben. Im Spezifischen wird dabei auf Freitextattribute (Kapitel 5), Meta-Attribute (Kapitel 6) und aggregierte Attribute (Kapitel 7) eingegangen.

In Kapitel 8 wird betrachtet, wie Verfahren zur Identifikation einzelner Attribute kombiniert werden können, um das Ziel der Dissertation, die Strukturierung kompletter Dokumente, zu erreichen. Zu diesem Zweck wird ein prototypisches Konzept zur Kombination der einzelnen Verfahren und dessen Evaluation im Rahmen einer Fallstudie in einer Domäne präsentiert. Die Dissertation wird in Kapitel 9 mit einer Zusammenfassung und einem Ausblick auf mögliche fortführende Arbeiten geschlossen.

2

GRUNDLAGEN

Dieses Kapitel führt in die Grundlagen ein, die im Rahmen dieser Dissertation zum Einsatz kommen und auf denen die später vorgestellten eigenen Beiträge zur Strukturierung textueller Dokumente aufbauen. Zunächst wird das Themenfeld des *Information Retrievals* betrachtet. Zum einen werden in der vorliegenden Arbeit Methoden des Information Retrieval genutzt und zum anderen liefert die Arbeit durch die Strukturierung textueller Dokumenten einen Beitrag, der als Ausgangspunkt für weitergehende Verfahren des Information Retrieval genutzt werden kann. Im Anschluss wird auf den Bereich des *maschinellen Lernens* und auf Konzepte zur Evaluation von maschinellen Lernverfahren eingegangen, da sich einige der Ansätze zur Entwicklung der Verfahren im Rahmen dieser Dissertation eignen. Weiterhin liefert dieses Kapitel einen Überblick über das Feld des *Natural Language Processing* mit einem Fokus auf die Schritte zur Vorverarbeitung textueller Daten, welche im Rahmen dieser Dissertation zur Verarbeitung der in den Dokumenten enthaltenen Texte genutzt werden.

2.1 INFORMATION RETRIEVAL

Ein großer Anteil der weltweit verfügbaren Daten liegt in unstrukturierter Form vor. In unterschiedlichen Quellen wird dieser Anteil auf 80-85% geschätzt [19, 23, 96]. Während aus strukturierten Daten unter Verwendung von Abfragesprachen benötigte Informationen zielgerichtet erlangt werden können, existieren keine allgemein einsetzbaren Abfragesprachen zur Erlangung von Informationen aus unstrukturierten Daten. Dies legt nahe, dass Methoden benötigt werden, um diese Informationen zugänglich zu machen. Das Forschungsgebiet des *Information Retrieval* beschäftigt sich mit der Auffindung solcher Informationen:

> *„Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).“*[1] [97, Seite 1]

Cole [30] beschreibt, dass die konkreten Mittel zur Erfüllung eines Informationsbedarfs, im Gegensatz zu Mitteln zur Erfüllung anderer Bedürfnisse, für den bedürftigen Menschen häufig nicht bekannt sind (vergleiche Tabelle 1). Es besteht also ein abstrakter Informationsbedarf, für dessen Erfüllung die Mittel unbekannt sind. Information Retrieval Systeme haben das Ziel, die tatsächlichen Mittel zur Erfüllung des Bedarfs zu bestimmen.

Wie im vorigen Kapitel dargestellt, werden im Rahmen dieser Dissertation Verfahren entwickelt, um Dokumente von einer unstrukturierten Form in eine strukturierte

1 *„Informationsgewinnung ist das Finden von Material (meist Dokumente) in unstrukturierter Form (meist Text) aus einer großen Sammlung (meist auf Computern gespeichert), zur Befriedigung eines Informationsbedarfs.“* (freie Übersetzung durch den Autor)

Tabelle 1: Menschliche Bedürfnisse und Mittel zur Erfüllung dieser (nach [30])

Bedarf		Mittel		Ziel
Hunger	→	Nahrungsaufnahme	→	Sättigung
Durst	→	Trinken	→	kein Durst
Informationsbedarf	→	*häufig unbekannt*	→	Erfüllung des Informationsbedarfs

Form zu überführen. Die strukturierte Form soll dabei die relevanten Informationen der ursprünglichen Dokumente enthalten. Dies erfordert die Identifikation der relevanten Informationen in den unstrukturierten Dokumenten. Dieser Informationsbedarf soll mittels Methoden des Information Retrievals erfüllt werden. In dieser Arbeit werden insbesondere Verfahren des maschinellen Lernens genutzt, um die Informationen von Interesse zu identifizieren. Maschinelles Lernen erlaubt das automatisierte Erkennen von Zusammenhängen in Daten. Somit wird keine explizite, arbeitsintensiv manuell zu erstellende Modellierung der Zusammenhänge benötigt.

2.2 MASCHINELLES LERNEN

Eine allgemein anerkannte Definition für *maschinelles Lernen* stammt vom Tom M. Mitchell [110, Seite 2]:

> *„A computer program is said to learn from experience* E
> *with respect to some class of tasks* T *and performance measure* P,
> *if its performance at tasks in* T, *as measured by* P,
> *improves with experience* E"[2]

Als *Task* ist eine konkrete Aufgabe zu sehen. Häufig sind dies vom Menschen erfüllbare Aufgaben, wie beispielsweise das Transkribieren gesprochener Worte, das Fahren eines Fahrzeuges oder das erfolgreiche Spielen eines Spiels. Als *Experience* (Erfahrung) werden in dieser Definition vorgegebene Beispiele oder Erfahrungen aus der Anwendung eines Ansatzes verstanden. Beispiele im Fall des Transkribierens gesprochener Worte sind Paare von Audioaufzeichnungen und dem Transkript der gesprochenen Worte. Erfahrungen aus der Anwendung im Falle des Spielens des Spiels sind die Ergebnisse des Spiels. *Performance Measures* sind Gütekriterien, mit denen bewertet wird, wie gut eine Aufgabe erfüllt wurde. Im Falle des Transkribierens wäre dies beispielsweise der Anteil der korrekt transkribierten Worte oder im Fall des Spielens des Spiels der Anteil gewonnener Spiele.

Verfahren des maschinellen Lernens lassen sich in drei Klassen zusammenfassen [138]. Beim *überwachten Lernen*[3] stehen dem Verfahren Beispiele zusammen mit ihren Aufgabenlösungen zur Verfügung. Dies können zum Beispiel Paare von Audioaufzeichnungen und dem Transkript der gesprochenen Worte oder Texte zusammen

2 *„Ein Computerprogramm wird als lernend aus einer Erfahrung* E *in Bezug auf eine Klasse von Aufgaben* T *und ein Performanzmaß* P *bezeichnet, wenn dessen Performanz bei Aufgaben* T, *gemessen durch* P, *sich durch Erfahrung* E *verbessert."* (freie Übersetzung durch den Autor)

3 engl. *Supervised Learning*

mit dem Thema des Textes sein. Beim *unüberwachten Lernen*[4] stehen dem Verfahren nur Beispiele ohne Aufgabenlösung zur Verfügung, dies könnten beispielsweise Wetterdaten für einzelne Tage sein, mit der Aufgabe, diese in verschiedene Typen von Tagen zu gruppieren. Hierbei wäre eine Gruppierung in „Tage mit gutem Wetter“ und „Tage mit schlechtem Wetter“ sinnvoll. Es können jedoch auch alternative, für den Menschen nicht direkt ersichtliche, Zusammenhänge als relevant identifiziert werden. Beim *bestärkenden Lernen*[5] lernt das System die Güte einer Handlungssequenz auf Basis von „Belohnungen“ oder „Bestrafungen“ für die gesamte Sequenz. Beispielsweise beim Spielen eines Spiels, welches aus einer Sequenz von Spielzügen besteht, wäre der Sieg eines Spiels eine Belohnung, während das Verlieren eine Bestrafung wäre. Häufig wird außerdem noch die Klasse des *semi-überwachten Lernens*[6] genannt, bei der sowohl Beispiele mit Aufgabenlösung als auch Beispiele ohne Aufgabenlösung zur Verfügung stehen.

2.2.1 *Klassifikation*

In der vorliegenden Arbeit werden vorrangig Verfahren des überwachten Lernens verwendet, da es bei diesen ermöglicht wird, die Zusammenhänge von Interesse durch Beispieldaten mit Aufgabenlösungen implizit festzulegen. Formell ist die Aufgabe des überwachten Lernens wie folgt definiert [138, Seite 695]:

> *„Given a training set of* N *example input-output pairs* $(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N)$ *where each* y_i *was generated by an unknown function* $y = f(x_i)$*, discover a function* h *that approximates the true function* f*.“*[7]

Es ist anzumerken, dass die Werte x und y nicht numerischer Natur sein müssen. Die Funktion h wird auch als *Hypothese* bezeichnet. Somit ist das Ziel des überwachten Lernens, aus der Anzahl der möglichen Hypothesen die Hypothese zu finden, welche am besten den Zusammenhang zwischen allen möglichen x_i und y_i beschreibt.

Wenn der Wertebereich der Funktion f, also die Menge aller möglichen y_i, eine vordefinierte endliche Menge bildet, spricht man von *Klassifikation*. Hierbei kann unterschieden werden zwischen *binärer Klassifikation*, bei der der Wertebereich nur aus zwei Werten besteht, und *multinominaler Klassifikation*, bei der der Wertebereich aus mehr als zwei Werten besteht. Weiterhin ist zwischen Klassifikationsproblemen, bei denen jede Instanz genau einer Klasse zuzuordnen ist (*Single-Label Klassifikation*) und Problemen, bei der eine Instanz mehreren Klassen gleichzeitig zugeordnet werden kann (*Multi-Label Klassifikation*), zu unterscheiden.

Um eine Eingabeinstanz x, wie beispielsweise ein Bild, ein Textdokument oder ein Wetterdatensatz, zu klassifizieren, muss diese für die meisten Klassifikationsverfahren in Form eines *Merkmalsvektors* repräsentiert werden. *Merkmale* erlauben es, eine Instanz abstrakt zu beschreiben. So könnten beispielsweise für die Klassifikation eines Bildes die einzelnen Farbanteile relevante Merkmale sein, oder für die Klassifikation eines Textes die darin enthaltenen Worte. Die relevantesten Merkmalstypen sind

4 engl. *Unsupervised Learning*
5 engl. *Reinforcement Learning*
6 engl. *Semi-Supervised Learning*
7 *„Gegeben sei eine Trainingsmenge von* N *beispielhaften Eingabe-Ausgabe-Paaren* (x_1, y_1), (x_2, y_2),...,(x_N, y_N) *bei der jedes* y_i *durch eine unbekannte Funktion* $y = f(x_i)$ *erzeugt wurde. Ziel ist die Suche nach einer Funktion* h*, die die tatsächliche Funktion* f *approximiert.“* (freie Übersetzung durch den Autor)

numerische Merkmale und *nominale Merkmale*. Numerische Merkmale können Gleitkommawerte oder ganzzahlige Werte annehmen, nominale Merkmale können Werte aus einer vordefinierten, endlichen Menge von Werten annehmen. [174]

Das Klassifikationsergebnis, also die Aussage, ob eine Eingabeinstanz einer Klasse zuzuordnen ist, kann numerischer Natur sein. In diesem Fall wird meist ein normalisierter Wert im Intervall [-1,0;1,0] ausgegeben, wobei Werte < 0 einer negativen Klassifikationsentscheidung entsprechen und Werte > 0 einer positiven Klassifikationsentscheidung. Dieser numerische Wert wird auch als *Konfidenzwert* des Klassifikators bezeichnet. Das Klassifikationsergebnis kann aber auch einen diskreten Wert annehmen, beispielsweise einen booleschen Wert oder Werte einer weiteren beliebig definierten Menge.

Das Bereitstellen der beispielhaften Eingabe-Ausgabe-Paare und das Bilden eines Klassifikationsmodells auf Basis dieser wird als *Training* des Klassifikators bezeichnet. Die Eingabe-Ausgabe-Paare werden *Trainingsinstanzen* genannt. Die Anwendung des Klassifikators auf neue zu klassifizierende Elemente wird als *Testen* bezeichnet. Abbildung 1 zeigt schematisch den Ablauf des Trainings und des Testens am Beispiel der Klassifikation von Dokumenten. Zunächst müssen die annotierten Trainingsdokumente in eine vektorisierte Form (den Merkmalsvektor) überführt werden, bevor sie zum Training des Klassifikators genutzt werden können. Nachdem der Klassifikator trainiert ist, können unbekannte Testdokumente, die auch zunächst in vektorisierte Form überführt werden müssen, vom Klassifikator klassifiziert werden.

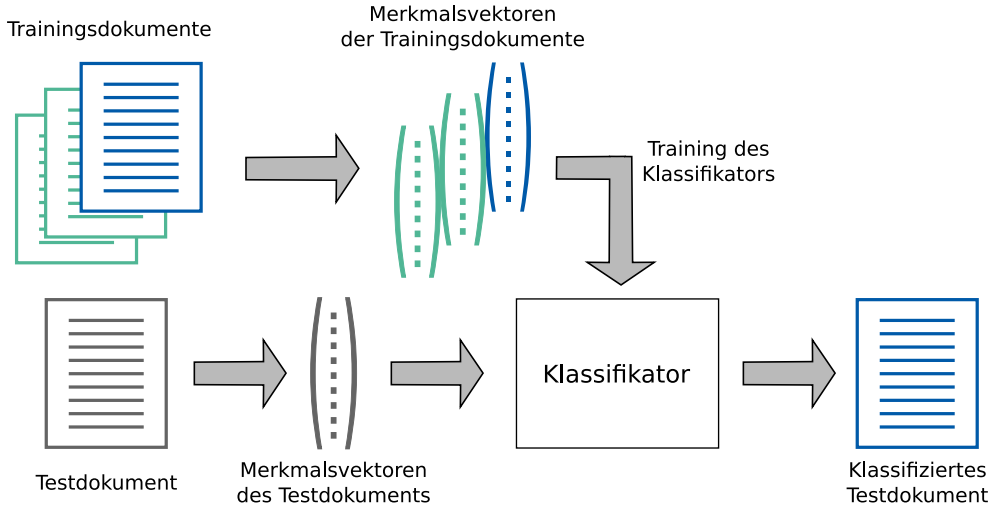


Abbildung 1: Schematischer Ablauf des Trainings und Testens eines Klassifikators am Beispiel einer Textklassifikation, die unterschiedlichen Farben repräsentieren beispielhafte Klassen

Die Annotation der Trainingsinstanzen muss häufig manuell erfolgen, da die Klasse der Instanzen nicht vorab bekannt ist. Die Erstellung der Annotationen ist sehr arbeitsintensiv, daher ist es erstrebenswert, möglichst wenige Trainingsinstanzen für das Training des Klassifikators zu benötigen.

Der Klassifikator selbst kann auf unterschiedlichen Modellen basieren. *Entscheidungsbäume* werden beim Testen startend an der Wurzel traversiert und bei jedem Knoten findet die Entscheidung bezüglich des weiteren Pfads abhängig von Merkmalswerten der zu klassifizierenden Instanz statt. Durch Erreichen eines Blattes wird die Zielklasse entschieden. Es existieren zahlreiche Ansätze, wie Entscheidungsbäu-

me auf Basis von Trainingsdaten konstruiert werden können [136]. Entscheidungsbäume haben den Vorteil, dass sie leicht vom Menschen nachvollziehbar sind.

Bayes-Klassifikatoren ordnen auf Basis des Satzes von Bayes [15] zu klassifizierende Instanzen der Klasse zu, bei der die Wahrscheinlichkeit zur Zugehörigkeit am größten ist. Sie zeichnen sich insbesondere durch eine einfache und schnelle Modellbildung aus. Ein populärer Vertreter der Bayes-Klassifikatoren ist der *Naive Bayes Klassifikator* [137].

Support Vector Machines (SVMs) nutzen das Modell eines hochdimensionalen Raumes, bei dem die einzelnen Instanzen Vektoren im Raum sind. Die Dimensionalität des Raumes definiert sich durch die Anzahl der Merkmale im Merkmalsvektor. Beim Trainieren wird eine trennende Hyperebene zwischen den Merkmalsvektoren der einzelnen Klassen identifiziert, wobei dabei der Abstand der Ebene zu den Vektoren maximiert wird [151]. SVMs haben insbesondere bei komplexen Klassifikationsproblemen eine hohe Klassifikationsgüte [77]. Hierzu zählen aufgrund der hohen Dimensionalität auch Textklassifikationsprobleme.

Ein künstliches *neuronales Netz* besteht aus mehreren Schichten mit jeweils mehreren parallel geschalteten Neuronen, welche das Eingangssignal aggregieren und ab einem Schwellwert der Aggregation ein Signal in die nächste Schicht weiterleiten [15]. Eine Herausforderung bei der Nutzung neuronaler Netze ist die Wahl der Anzahl der Schichten, der Aggregationsfunktion und der Schwellwerte [174].

Unter Verwendung von *Conditional Random Fields (CRF)* werden nicht nur einzelne Instanzen unabhängig voneinander klassifiziert, sondern den einzelnen Instanzen in einer Sequenz von Instanzen eine Klasse zugeordnet [85]. Dabei werden insbesondere auch die vorhergesagten Klassen für andere Instanzen in der Sequenz berücksichtigt, so lässt sich die Vorhersage der vollständigen Klassensequenz optimieren [165].

Für alle Klassifikationsansätze gilt, dass es einen starken Zusammenhang zwischen der Anzahl der Trainingsinstanzen und der Güte der Klassifikation gibt. Dieser Zusammenhang ist beispielhaft für zwei Klassifikatoren in Abbildung 2 in Form der *Lernkurven* dargestellt.

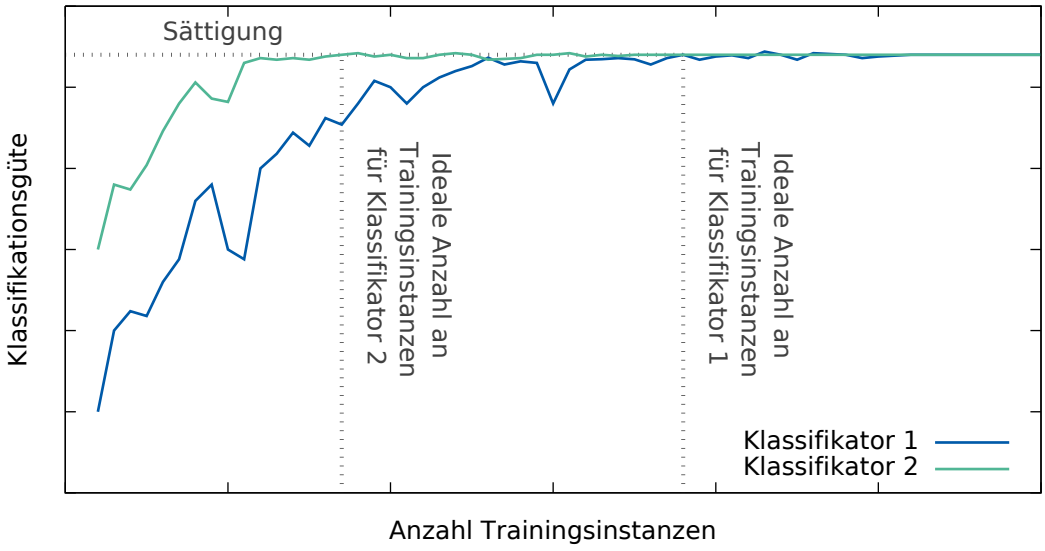


Abbildung 2: Typischer Verlauf einer Lernkurve (nach [98])

Mit steigender Anzahl an Trainingsinstanzen steigt die Klassifikationsgüte bis hin zu einer Sättigung. Weiterhin ist gerade bei kleinen Anzahlen an Trainingsinstanzen eine starke Unstetigkeit der Lernkurve zu erkennen. Ab dem Sättigungspunkt führt das weitere Hinzunehmen von Trainingsinstanzen zu keiner weiteren Verbesserung der Klassifikationsgüte. Idealerweise würde man genau diese Anzahl an Instanzen zum Training zur Verfügung stellen, da so zum einen die Menge der zu annotierenden Instanzen gering gehalten werden kann und zum anderen die Bildung des Klassifikationsmodells bei kleinerer Anzahl an Trainingsinstanzen weniger rechenintensiv ist. Die Bestimmung der Sättigung ist jedoch nicht trivial, da es im Normalfall auch bei großer Zahl an Trainingsinstanzen noch zu (geringen) Schwankungen in der Klassifikationsgüte kommt und weiterhin die eintretende Sättigung erst dann erkannt werden kann, wenn bereits mehr als die benötige Menge an Trainingsinstanzen zum Training verwendet wurden. Klassifikator 2 wäre im dargestellten Beispiel Klassifikator 1 vorzuziehen, da bereits bei weniger Trainingsinstanzen eine Sättigung gleicher Höhe eintritt. Sowohl die Steigung der Lernkurve als auch die Höhe der Sättigung hängt vom verwendeten Klassifikator und von der Wahl der Merkmale ab. Weiterhin hat das betrachtete Klassifikationsproblem einen wesentlichen Einfluss auf diese beiden Charakteristika.

2.2.2 *Textklassifikation*

Bei der Klassifikation von Texten können unterschiedliche Aspekte als Zielklassen verwendet werden. Sehr häufig ist das Thema eines Textes Klassifikationskriterium. Ein Text kann auch anhand anderer Kriterien wie der Sprache des Textes, des Textgenres oder der emotionalen Haltung des Autors zum Geschriebenen klassifiziert werden. Ein weiteres Anwendungsbeispiel der Textklassifikation ist die Klassifikation von E-Mails in die Klassen „Spam“ und „kein Spam“.

Zur Klassifikation von Texten können verschiedene die Texte repräsentierende Merkmale verwendet werden. Am häufigsten werden *Unigramme* verwendet. Unigramme sind die in den Texten vorkommenden Zeichenketten, die durch Separatoren wie Leerzeichen voneinander getrennt sind [153]. Beispiele für Unigramme sind „Hund“, „ist“, „Europäische“. Auch die verallgemeinerte Form, *n-Gramme*, die auch als *Phrasen* bezeichnet werden, wird eingesetzt [53]. N-Gramme sind die Konkatenation n getrennter Zeichenketten [27], Beispiele für *2-Gramme* beziehungsweise *Bigramme* sind „wir gehen“, „Europäische Union“, „die in“. Unigramme und n-Gramme bieten sich insbesondere zur thematischen Klassifizierung von Texten an. Um Merkmalsvektoren für ein Klassifikationsproblem nutzen zu können, müssen sie von gleicher Dimensionalität sein, somit muss für ein Textklassifikationsproblem zunächst die Menge der verwendeten Unigramme oder n-Gramme definiert werden. Dies geschieht meist auf Basis der Trainingsdokumente, das heißt nur Unigramme beziehungsweise n-Gramme, die in den Trainingsdokumenten auftauchen, können in Merkmalsvektoren verwendet werden. Häufig wird die Anzahl der Merkmale auf die in den Trainingsdokumenten am häufigsten vorkommenen Unigramme begrenzt und sehr seltene Unigramme werden nicht verwendet.

Als weitere Merkmale zur Klassifikation von Texten können beispielsweise die Häufigkeit von Abkürzungen, Umgangssprache, Auftauchen von zeitlichen Indikatoren oder Metadaten-basierte Merkmale genutzt werden [160]. Für die Klassifikati-

on des Textgenres können auch stylistische Merkmale, wie die Länge eines Textes, die Anzahl der Wörter pro Satz oder die Anzahl von Satzzeichen relevant sein [50].

Bei der Verwendung von Unigrammen als Merkmale spricht man auch vom *Bag of Words Modell*. Die Reihenfolge, in der die einzelnen Unigramme im Text auftauchen, hat keinen Einfluss auf die Bildung des Merkmalsvektors. Im einfachsten Fall sind die Merkmale binär, es wird also nur betrachtet, ob ein Wort in einem Dokument auftaucht oder nicht. Bessere Klassifikationsergebnisse lassen sich erzielen bei Verwendung von Häufigkeiten der einzelnen Worte als Merkmale, da davon ausgegangen wird, dass im Text häufiger auftretende Begriffe eine höhere Relevanz für diesen Text haben [95]. Eine Voraussetzung für die Verwendung von Merkmalen zur Unterscheidung zwischen einzelnen Klassen ist eine Varianz der Merkmalswerte für die einzelnen Instanzen. Wenn alle Instanzen für ein spezifisches Merkmal einen hohen oder niedrigen Wert haben, so kann dies nicht für die Klassifikation genutzt werden, da anhand dieses Wertes keine Klassifikationsentscheidung getroffen werden kann. Der *Informationsgewinn*[8] [110] ist in diesem Fall gering. Daher sind Terme, die in allen Dokumenten häufig vorkommen, von geringer Relevanz. Auf Basis dieser Beobachtung hat sich die Verwendung des *Term Frequency - Inverse Document Frequency (TF-IDF)* Maßes etabliert [97]. Dies wird berechnet durch [159]

$$\text{tf-idf}(t_i, d_j, D) = tf_{t_i,d_j} * \log(\frac{|T|}{n_i}), \tag{1}$$

wobei der erste Faktor die Termfrequenz (*tf*) und der zweite Faktor die inverse Dokumentenfrequenz (*idf*) angibt. Es gelten folgende Definitionen:

t_i	i-te Term im Merkmalsvektor
d_j	j-te Dokument im Dokumentenkorpus D
D	Dokumentenkorpus
T	Trainingskorpus ($T \subset D$)
tf_{t_i,d_j}	Häufigkeit des Vorkommens von Term t_i in Dokument d_j
n_i	Anzahl der Dokumente in T in denen Term t_i auftaucht.

Die TF-IDF Gewichtung bewirkt, dass Terme, die häufig in einem Dokument vorkommen, aber relativ selten in der Gesamtheit der Dokumente eines Dokumentenkorpus vorkommen, durch einen relativ hohen Merkmalswert repräsentiert werden.

Sowohl unter Betrachtung der Charakteristika von Textklassifikationsproblemen als auch in experimentellen Versuchen konnte gezeigt werden, dass SVMs aufgrund ihres Aufbaus sehr geeignet zur Textklassifikation und anderen Klassifikationsansätzen häufig überlegen sind [73, 75, 153].

Für weitergehende Betrachtungen zur automatisierten Klassifikation von Texten wird an dieser Stelle auf entsprechende Übersichtsartikel verwiesen [2, 153].

2.2.3 *Evaluation*

Zur Beurteilung von Klassifikationsverfahren sollte ein standardisierter Prozess durchgeführt werden. Dies erfordert zunächst einen *Evaluationskorpus* (Abschnitt 2.2.3.1). Weiterhin werden aussagekräftige Maße zur Beschreibung der Güte der Klassifikationsverfahren benötigt (Abschnitt 2.2.3.2). Um statistisch aussagekräftige Evaluations-

8 engl. *Information gain*

ergebnisse zu erhalten, sollte ein standardisiertes Evaluationsverfahren (Abschnitt 2.2.3.2) angewandt werden.

2.2.3.1 *Evaluationskorpora*

Evaluationskorpora, bestehend aus zu klassifizierenden Instanzen zusammen mit ihren tatsächlichen Klassenzugehörigkeiten (*„Label“*), werden sowohl zum Training eines Klassifikators als auch zum Testen eines solchen benötigt. Bei der Auswahl der Korpora sollte beachtet werden, dass diese repräsentativ für das Anwendungsszenario sind, in dem der Klassifikator verwendet werden soll. Voraussetzungen für diese Repräsentativität sind sowohl eine gleiche Klassenverteilung, also ein gleiches Verhältnis von positiven zu negativen Instanzen wie im Anwendungsszenario, als auch eine gleiche Verteilung der Merkmalswerte [122]. Dies kann aufgrund der Unterschiedlichkeit der Anwendungsszenarien meist nur durch Verwenden von Instanzen direkt aus dem Anwendungsszenario heraus erreicht werden. Im Kontrast dazu steht der Bedarf an Standardkorpora: Klassifikationsprobleme sind kaum miteinander vergleichbar, so dass ein Klassifikator mitunter bei Verwendung unterschiedlicher Datensätze eine stark variierende Klassifikationsgüte aufweisen kann. Um die Vergleichbarkeit zwischen Klassifikationsverfahren zu gewährleisten, sollten daher verschiedene Verfahren unter Verwendung einheitlicher Datensätze evaluiert werden. Hierzu werden *Goldstandard*-Datensätze [97] verwendet.

2.2.3.2 *Evaluationsmaße*

Im Rahmen dieser Arbeit werden verschiedene etablierte Maße zur Beschreibung der Güte der vorgestellten Verfahren eingesetzt (vergleiche [97]). Die einzelnen Maße haben unterschiedliche Zielsetzungen, sie basieren jedoch alle auf den Werten, die sich aus einer *Konfusionsmatrix* ableiten lassen. Eine solche Matrix für eine binäre Klassifikation, bei der Instanzen als „positiv“ oder als „negativ“ klassifiziert werden können, ist in Tabelle 2 dargestellt. Während die einzelnen Zeilen für die tatsächliche Klasse einer Instanz stehen, stehen die Spalten für die Ergebnisse der Klassifikation durch den Klassifikator. Die ganzzahligen Werte TP, FN, FP und TN spiegeln die Anzahl der Instanzen der jeweiligen Kategorie wider. FP stellt beispielsweise die Anzahl der Instanzen dar, die als „negativ“ hätten klassifiziert werden sollen, aber als „positiv“ klassifiziert wurden. Die Summe der Werte beträgt immer N, welches die Gesamtzahl der zu klassifizierenden Instanzen ist. Idealerweise gilt $FP = FN = 0$, da in diesem Fall alle Instanzen korrekt klassifiziert wurden. [157]

Tabelle 2: Konfusionsmatrix (TP = *„True Positive“* = korrekterweise als positiv klassifiziert; FN = *„False Negative“* = fälschlicherweise als negativ klassifiziert; FP = *„False Positive“* = fälschlicherweise als positiv klassifiziert; TN = *„True Negative“* = korrekterweise als negativ klassifiziert)

		Klassifiziert als		
		positiv	*negativ*	Summe
Tatsächliche Klasse	*positiv*	TP	FN	$TP + FN$
	negativ	FP	TN	$FP + TN$
	Summe	$TP + FP$	$FN + TN$	N

Die *Accuracy*[9] eines Klassifikators für ein Klassifikationsproblem gibt an, welcher Anteil der zu klassifizierenden Instanzen korrekt klassifiziert wurde. Hierbei wird nicht berücksichtigt, ob die Instanzen positiv oder negativ sind.

$$\mathrm{Accuracy} = \frac{TP + TN}{N} \tag{2}$$

Mit der *Precision*[10] eines Klassifikators wird angegeben, welcher Anteil der als positiv klassifizierten Instanzen tatsächlich positiv ist. Somit ergibt sich folgende Definition:

$$\mathrm{Precision} = \frac{TP}{TP + FP}. \tag{3}$$

Der *Recall*[11] beschreibt, welcher Anteil der tatsächlich positiven Instanzen vom Klassifikator als positiv klassifiziert wurde:

$$\mathrm{Recall} = \frac{TP}{TP + FN}. \tag{4}$$

Bei der Bildung eines Klassifikationsmodells kann die Tendenz des Klassifikators zugunsten des Precision- oder des Recall-Wertes beeinflusst werden, was in Abhängigkeit der Anwendung sinnvoll sein kann. So ist beispielsweise für einen Suchenden, der Webseiten, die Informationen über Grippesymptome enthalten, finden will, ärgerlich, wenn viele der gefundenen Webseiten keine Grippesymptome beschreiben. In diesem Fall wäre eine hohe Precision erstrebenswert. In diesem Beispiel wäre es weniger wichtig, dass alle Webseiten, die Grippesymptome beschreiben, korrekt als positiv klassifiziert werden würden. Wenn jedoch ein Suchender im Internet Informationen über ein spezifisches, seltenes Grippesymptom finden möchte, nimmt er es eher in Kauf, auch viele irrelevante Seiten zu betrachten, solange die gesuchte Information darunter ist. In diesem Falle wäre eher ein hoher Precisionwert erstrebenswert.

Wenn im Extremfall alle Instanzen als positiv klassifiziert werden, ergibt sich ein Recall von 1,0, wohingegen die Precision dem Anteil der positiven Instanzen aus der Gesamtmenge der zu klassifizierenden Instanzen entspricht. Gleiches gilt im umgekehrten Fall, wenn keine Instanz als positiv klassifiziert wurde. Um mit einem einzelnen Maß allgemeingültigere Aussagen über die Klassifikationsgüte treffen zu können, wird häufig das harmonische Mittel von Precision und Recall verwendet, das F_1-*Maß*:

$$F_1\text{-Maß} = \frac{2 * \mathrm{Precision} * \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}. \tag{5}$$

Die beschriebenen Maße lassen sich auch zur Evaluation eines Multi-Klassen-Klassifikators verwenden. Dies kann per *macro-averaging* oder *micro-averaging* erfolgen. Beim macro-averaging wird zunächst für jede Klasse eine eigene Konfusionsmatrix erstellt, in der die Klassifikation „positiv" der Zugehörigkeit zur Klasse entspricht, während die Klassifikation „negativ" repräsentiert, dass die Instanz nicht zur Klasse gehört. Auf Basis dieser Konfusionsmatrix kann das entsprechende Maß für diese Klasse bestimmt und das arithmetische Mittel der Werte für die einzelnen Klassen

9 Zur vereinfachten Lesbarkeit werden in dieser Arbeit die etablierten englischsprachigen Begriffe für Evaluationsmaße verwendet. In der deutschsprachigen Literatur wird die Accuracy auch als *Korrektklassifikationsrate* bezeichnet [137].

10 deutsch *Genauigkeit* [137]

11 deutsch *Sensitivität* oder auch *Richtig-Positiv-Rate* [137]

gebildet werden. Beim micro-averaging hingegen werden die kumulativen Werte für TP, FN, TN und FP über alle Klassen hinweg betrachtet und auf Basis der entstehenden kumulativen Konfusionsmatrix der Wert des zu bestimmenden Maßes berechnet. Dieses Vorgehen bewirkt eine Gewichtung nach Klassengröße, wohingegen beim macro-averaging der Einfluss aller Klassen identisch ist. [157]

2.2.3.3 *Evaluationsverfahren*

Für die Evaluation von Klassifikationsverfahren ist es notwendig, dass sowohl zum Trainieren als auch zum Testen gelabelte Instanzen zur Verfügung stehen. Hierbei sollte eine strikte Trennung zwischen Training- und Testdaten erfolgen. Ein Testen unter Verwendung der Trainingsdaten würde kein realistisches Bild über die Klassifikationsgüte widerspiegeln [174]. Bei Verwendung eines Evaluationskorpus besteht bei Aufteilung dieses in Trainings- und Testdaten die Gefahr einer unrepräsentativen Aufteilung, bei der beispielsweise in den Trainingsdaten ein überdurchschnittlicher Anteil an Instanzen einer spezifischen Klasse enthalten ist. Weiterhin kann die Verteilung der Merkmalswerte variieren. Um diesen Effekt zu reduzieren, hat sich im maschinellen Lernen zur Evaluation das System der k-*fachen Kreuzvalidierung* etabliert (siehe Abbildung 3).

Bei der Kreuzvalidierung wird der Evaluationskorpus zunächst in k gleich große, zufällige Teilmengen aufgeteilt (entspricht den Spalten in der Abbildung). Anschließend werden k Evaluationsvorgänge durchlaufen (entspricht den Zeilen in der Abbildung), bei denen jeweils $k-1$ Teilmengen zum Trainieren verwendet werden und die k-te Teilmenge zum Testen verwendet wird. Für jeden der Vorgänge werden die gewählten Evaluationsmaße berechnet und abschließend wird das arithmetische Mittel der k Vorgänge gebildet. Bei der *stratifizierten Kreuzvalidierung* werden die Teilmengen nicht zufällig gebildet; die Aufteilung erfolgt unter Berücksichtigung der Klassenlabel, so dass in jeder der Teilmengen die gleiche Klassenverteilung zu finden ist wie in der Gesamtmenge. Kohavi [81] konnte zeigen, dass die repräsentativsten Ergebnisse unter Verwendung der stratifizierten k-fachen Kreuzvalidierung bei $k = 10$ erzielt werden.

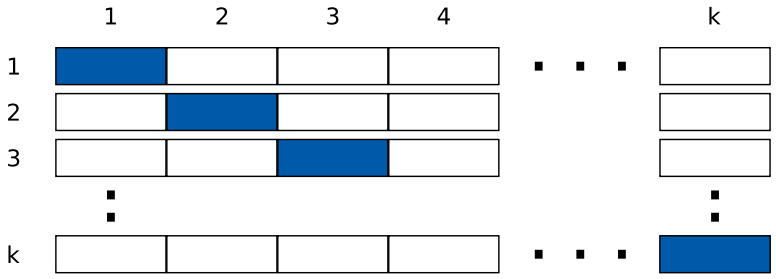


Abbildung 3: 10-fach Kreuzvalidierung (blau eingefärbte Rechtecke stellen die Testmenge dar, weiß gefärbte Rechtecke stellen die Trainingsmenge dar, jede Zeile repräsentiert einen Evaluationsdurchlauf)

2.3 NATURAL LANGUAGE PROCESSING

Das Ziel des *Natural Language Processing* ist das rechnerbasierte Bearbeiten von Aufgaben, in denen menschliche Sprache von Relevanz ist [75]. Dies trifft auch auf die in

dieser Dissertation adressierten Aufgaben zu. Weitere populäre Anwendungen sind die maschinelle Erstellung von Übersetzungen [80], Informationsextraktionsysteme [72], automatische Dialogsysteme [105] oder Systeme zur Zusammenfassung von Texten [116]. Als eine der größten Herausforderungen bei Aufgaben dieser Art wird die Ambiguität, also die Mehrdeutigkeit, der menschlichen Sprache gesehen [98]. So können beispielsweise Satzstruktur, Wortformen oder Wortbedeutungen mehrdeutig sein.

Für eine Vielzahl von Aufgaben des Natural Language Processing haben sich mehrere Vorverarbeitungsschritte als hilfreich oder notwendig herausgestellt. Die Schritte, die im Rahmen dieser Arbeit verwendet werden, sind im Folgenden dargestellt.

Das Zerteilen eines Textes in seine einzelnen *Token*, die atomaren Textbausteine, wird als *Tokenisierung* oder auch Wortsegmentierung bezeichnet. Häufig kann diese Trennung auf Basis der Leerzeichen durchgeführt werden. Diese leerzeichenbasierte Trennung reißt Token jedoch aus ihrem textuellen Kontext oder ändert ihre Bedeutung; die einzelnen Token in „New York“ oder „Angela Merkel“ verändern ihre Bedeutung durch Separierung, da „York“ auf eine Stadt in Großbritannien und „Merkel“ auf eine Stadt in Texas verweist. Weiterhin muss die Sprachabhängigkeit der Tokenisierung berücksichtigt werden (vergleiche zum Beispiel „house owner“ und „Hausbesitzer“). In manchen Fällen kann die Trennung an einem Sonderzeichen sinnvoll sein („I'm“ wird zur Tokenliste [„I“, „am“]), während davon in anderen Fällen abgesehen werden sollte („Check-in“). Auch die Behandlung von Sonderzeichen als Bestandteil eines Tokens kann variieren: während der Punkt an einem Satzende eher nicht als Bestandteil des vorhergehenden Tokens betrachtet werden sollte, sollte dies für Währungssymbole angewandt werden. [75]

Stemming ist der Prozess der Transformation eines Wortes zu seinen Wortstamm. So können durch den Prozess des Stemmings beispielsweise die Begriffe „fox“ und „foxes“ auf ihren Stamm „fox“ zurückgeführt werden. Dies erlaubt bei Textklassifikationsverfahren, dass alle Wörter mit dem gleichen Wortstamm durch ein gemeinsames Unigramm-Merkmal repräsentiert werden, wodurch sich häufig die Trefferqenauigkeit erhöhen lässt. Es kann jedoch auch zu unerwünschten Zuordnungen zwischen Wörtern kommen, so haben beispielsweise die englischen Wörter „customer“ und „custom“ den gemeinsamen Stamm „custom“[12], welcher eine wesentliche Sinnveränderung für das Wort „customer“ bedeutet. Häufig muss somit beim Stemming abgewogen werden, ob eher auf einen kurzen oder längeren Stamm zurückgeführt wird. Bei einem längeren Wortstamm können erwünschte Treffer zwischen verwandten Worten teils nicht erfolgen, während es bei zu kurzem Wortstamm eher zu Fehlern in der Zuordnung kommt. Auch Stemming-Verfahren sind stark sprachabhängig, da häufig explizite Regeln zur Rückführung auf den Wortstamm verwendet werden [94]. Im Gegensatz zum Stemming wird bei der *Lemmatisierung* versucht, ein Wort auf seine Grundform zurückzuführen. Dies kann insbesondere für Verben Vorteile bieten, da sie so auf ihren Infinitiv zurückgeführt werden können, was im Gegensatz zum Stemming die Zuordnung zwischen verschiedenen zeitlichen Formen eines unregelmäßig gebeugten Verbs erlaubt.

Unter *Part-of-Speech-Tagging (POS-Tagging)* versteht man die Annotation von Wörtern mit ihren jeweiligen Wortarten, wie beispielsweise Verben, Präpositionen oder

12 Ergebnis der Online-Demonstration des *Snowball Stemmers*, http://snowball.tartarus.org/demo.php, zugegriffen am 10.07.2015

Artikeln. Durch die begrenzte Anzahl der Wortarten stellt POS-Tagging ein Klassifikationsproblem dar, bei dem die Wortart die Klasse für die zu klassifizierenden Instanzen der Wörter darstellt. Die Herausforderung bei der Zuordnung liegt häufig in der Ambiguität der Wörter, so kann ein Wort von unterschiedlicher Wortart sein. Beispielsweise kann das Wort „run“ in der englischen Sprache sowohl ein Verb als auch ein Substantiv darstellen. Die konkrete Wortart definiert sich allein über den Kontext. Zur Bestimmung der Wortart werden sowohl regelbasierte Verfahren als auch maschinelle Lernverfahren eingesetzt [75]. Dabei werden unterschiedliche Mengen von Labeln verwendet, die die Wortarten repräsentieren. Im Rahmen dieser Arbeit wird für englischsprachige Dokumente die im „Penn Treebank Tagset“ [100] und für deutschsprachige Dokumente die im „Stuttgart-Tübingen-Tagset“ [145] definierte Menge an Labeln verwendet. Die Liste der Label in beiden Tagsets ist in Anhang A.1 zu finden. Tabelle 3 zeigt beispielhaft das Ergebnis einer Part-of-Speech (POS)-Annotation für einen englischsprachigen Satz unter Verwendung des Penn Treebank Tagset.

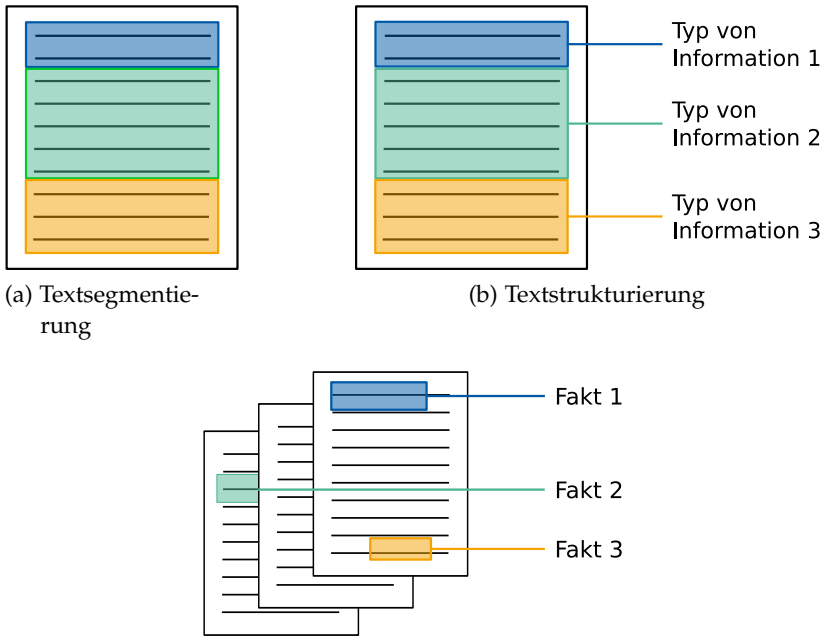
Tabelle 3: Beispiel für die POS-Annotation für den englischsprachigen Satz „Foxes are running on the street“

Token	POS-Tag	Beschreibung
Foxes	NNS	Substantiv, plural
are	VBP	Verb, nicht 3. Person, singular, Präsenz
running	VBG	Verb, Gerundium
on	IN	Präposition
the	DT	Artikel
street	NN	Substantiv, singular

3

VERWANDTE ARBEITEN

Die vorliegende Dissertation beschäftigt sich mit Methoden der Überführung unstrukturierter textueller Dokumente in eine strukturierte Form. Die strukturierte Form soll alle für eine Anwendung relevanten Informationen der Ausgangsdokumente enthalten. Die Zielsetzung des Verarbeitens von Texten zur Erkennung darin enthaltener relevanter Elemente zur weiteren Nutzung wurde in der Vergangenheit in zahlreichen anderen Arbeiten betrachtet. Der Fokus dieser Arbeiten variiert jedoch, eine Darstellung der unterschiedlichen Zielsetzungen ist Abbildung 4 zu entnehmen und wird im Folgenden beschrieben. Die farblich markierten Bereiche in der Abbildung stellen jeweils die adressierten strukturellen Elemente eines Textes dar.



(a) Textsegmentierung

(b) Textstrukturierung

(c) Informationsextraktion

Abbildung 4: Visualisierung der in diesem Kapitel beschriebenen Forschungsbereiche Textsegmentierung, Textstrukturierung und Informationsextraktion

Forschungsarbeiten zur *Segmentierung* von textuellen Dokumenten (Abbildung 4a) zielen darauf ab, Texte in sinnvolle Segmente zu unterteilen (beschrieben in Abschnitt 3.1). Bei den Methoden der Textsegmentierung bleibt jedoch die semantische Rolle des Segments für das Dokument, also der Typ der Information, unklar. Im Gegensatz dazu stehen Methoden der *Textstrukturierung* (Abbildung 4b), bei denen versucht wird, den einzelnen Segmenten auch einen Typ zuzuordnen, also einen Bezeichner, der den Inhalt des Elements beschreibt (beschrieben in Abschnitt 3.2). Das Ziel der *Informationsextraktion* (Abbildung 4c) wiederum ist die Identifikation von

einzelnen Informationen (Fakten) in Texten, die für konkrete Anwendungen von Interesse sind (beschrieben in Abschnitt 3.3). Die Herkunft der einzelnen Informationen ist meist irrelevant, daher ist die vollständige Abdeckung der einzelnen Texte nicht notwendig.

Die im Rahmen dieser Arbeit adressierten Methoden zur Strukturierung von Dokumenten sollen nicht auf spezifische Domänen angepasst sein, sondern mit möglichst geringem manuellen Aufwand in andere Domänen übertragen werden können. Zur Identifikation des Typs eines Segmentes mit dem Ziel der Textstrukturierung oder zum Ableiten von globalen Informationen über ein Dokument lassen sich Verfahren der Textklassifikation verwenden. Im Forschungsbereich der Textklassifikation wurden unterschiedliche domänenadaptive Verfahren vorgestellt, welche es erlauben, in einer neuen Domäne ohne großen manuellen Aufwand eingesetzt zu werden. Auf diese Verfahren wird in Abschnitt 3.4 eingegangen.

3.1 TEXTSEGMENTIERUNG

Um Verfahren der *Textsegmentierung* vorstellen zu können, ist zunächst eine Definition von *Textsegmenten* notwendig. Kozima [84, Seite 1] beschreibt Textsegmente wie folgt:

> *„A text segment, whether or not it is explicitly marked, as are sentences and pagragraphs, is defined as a sequence of clauses or sentences that display local coherence."*[1]

Daran angelehnt definiert Scholl [152, Seite 91] Segmente in der Textgattung der Webseiten als

> *„strukturell unabhängige, kohärente Blöcke mit unterschiedlicher Funktionalität"*[2].

In beiden Definitionen bezieht sich die Kohärenz auf die Semantik der Segmente. Während innerhalb eines Segments eine hohe semantische Kohärenz besteht, besteht zwischen mehreren Segmenten eine niedrige Kohärenz. Aus diesen Definitionen der Textsegmente leitet sich die Aufgabe der Textsegmentierung ab als die Identifikation der Stellen im Text, an denen ein Segment endet und das folgende Segment beginnt [12]. Dies entspricht der Identifikation thematischer Grenzen in Texten [29].

Die Ansätze zur Segmentierung lassen sich in zwei Klassen aufteilen. Die erste Klasse umfasst Ansätze, welche das Layout des Textes nutzen, um eine Segmentierung vorzunehmen (beschrieben in Abschnitt 3.1.1). Es wird davon ausgegangen, dass sich auf Basis des Layouts eines Textes dessen Segmente identifizieren lassen. Die Verfahren dieser Klasse sind fokussiert auf die Segmentierung von Webseiten. Eine Segmentierung von Rohtext ist mit diesen Verfahren nicht möglich. Die zweite Klasse umfasst Ansätze, welche auf Basis des Inhalts der Texte eine Segmentierung vornehmen (beschrieben in Abschnitt 3.1.2). Im Folgenden wird ein Überblick über Verfahren der zwei Klassen gegeben. Eine eindeutige Kategorisierung ist nicht immer möglich. Bei Uneindeutigkeiten wird eine Kategorisierung in die dominierende Klasse des Ansatzes vorgenommen.

1 *„Ein Textsegment, egal ob explizit markiert, wie es Sätze oder Absätze sind, oder nicht markiert, ist definiert als eine Sequenz von Klauseln oder Sätzen, die eine lokale Kohärenz aufweisen"* (freie Übersetzung durch den Autor)

2 freie Übersetzung durch den Autor, die originale Definition lautet *„[...] structurally independent, coherent blocks that serve different functionalities."*

3.1.1 *Layoutbasierte Segmentierung von Webseiten*

Bei der Verwendung des Layouts zur Segmentierung können sowohl die interne Repräsentation des Dokuments mittels einer Markup-Sprache, wie beispielsweise die *Hypertext Markup Language (HTML)*, als auch das tatsächliche visuelle Layout der vom Internetbrowser gerenderten Webseite verwendet werden [152].

Im Verfahren *Block Fusion* [82] wird eine Segmentierung auf Basis von Schwankungen der textuellen Dichte in den HTML-Repräsentationen der Webseiten durchgeführt. Dafür wird betrachtet, wie viele textuelle Token, die keine HTML-Elemente darstellen, pro Zeile im HTML-Dokument vorliegen. Wenn die Differenz der textuellen Dichte zwischen zwei Zeilen groß ist, wird eine Segmentgrenze angenommen. Darüber hinaus kommen Optimierungen zum Einsatz, welche Glättungen der Dichtevarianz vornehmen, um inkorrekte Segmentgrenzen durch eine zu hohe Sensitivität zu verhindern. Grundlage für das Verfahren ist die Annahme, dass in einem Segment eine homogene Struktur vorliegt und daher die textuelle Dichte nur geringen Schwankungen unterliegt. Bei einer Segmentgrenze wiederum tauchen vermehrt Markup-Elemente auf, die Überschriften oder Abstände im Text einführen können. Auch wenn beispielsweise von einer Auflistung zu einem Fließtext gewechselt wird, ändert sich die Dichte, da in der Aufzählung viele Markup-Elemente vorliegen, während dies im Fließtext eher wenige sind. Diese Definition der Segmentgrenze entspricht einer Änderung des visuellen Erscheinungsbild an der entsprechenden Stelle der gerenderten Webseite.

Eine Segmentierung direkt auf Basis des visuellen Erscheinungsbildes wird durch das Verfahren *Vision-based Web Page Segmentation (VIPS)* [25] vorgenommen. Hierbei wird zunächst unter Verwendung visueller Merkmale wie Hintergrundfarbe oder Schriftgröße eine Aufteilung der Webseite in Blöcke vorgenommen. Auf Basis dieser Blöcke werden initiale Segmentgrenzen zwischen den Blöcken definiert. Das Gewicht der Grenzen wird auf Basis der Unterschiede der benachbarten Blöcke definiert. Unter Verwendung der Segmentgrenzen werden nun Blöcke, die von Segmentgrenzen mit geringem Gewicht begrenzt werden, wieder zusammengeführt. Nun wird für jedes der Segmente, also einer Menge von Blöcken, ein Kohärenzmaß auf Basis des visuellen Erscheinungsbilds bestimmt. Falls das Kohärenzmaß einen Grenzwert unterschreitet, wird das Segment erneut unter Verwendung visueller Merkmale in Blöcke unterteilt, welche neue Segmentgrenzen definieren. Dieser Prozess wird rekursiv durchlaufen bis das Kohärenzmaß aller Segmente den Grenzwert überschreitet und somit von der Atomarität der Segmente ausgegangen werden kann.

Eine Kombination der Nutzung der HTML-Repräsentation und der visuellen Erscheinung wird im Verfahren *Block-o-Matic* [140] zur Segmentierung verwendet. Die HTML-Repräsentation wird in eine Repräsentation gemäß des *Document Object Model (DOM)* [101] überführt. Die einzelne Elemente des DOM werden anschließend unter Verwendung geometrischer Kriterien, wie einer minimalen Segmentgröße, auf Basis der gerenderten Repräsentation zusammengefügt.

Die vorgestellten Verfahren Block Fusion, VIPS und Block-o-Matic wurden von Sanoja und Gançrski [141] unter einheitlichen Bedingungen evaluiert. Im arithmetischen Mittel des Prozentsatzes der korrekten Segmentierungen über fünf Kategorien von Webseiten (Blogs, Foren, Bilder-Webseiten, Unternehmenswebseiten und Wikis) wurden die besten Ergebnisse durch den hybriden Block-o-Matic-Ansatz erreicht.

Die schlechtesten Ergebnisse wurden für Block Fusion erreicht, während VIPS eine mittlere Qualität erreichte. Bei Betrachtung der prozentualen Abdeckung der Texte schnitt Block-o-Matic schlechter als VIPS ab.

Layoutbasierte Verfahren benötigen die Verwendung spezifischer Dokumentenformate wie HTML zur Segmentierung. Eine Segmentierung von Rohtext ist nicht möglich. Weiterhin basieren die Verfahren auf der Annahme, dass ein neues Segment durch eine Änderung des visuellen Erscheinungsbilds eingeleitet wird. Somit sind Segmente mit gleicher Struktur, zwischen denen keine visuelle Trennung vorliegt oder die keinen visuellen Unterschied haben, nicht segmentierbar.

3.1.2 *Inhaltsbasierte Segmentierung*

Das maßgebliche Kriterium zur Definition der Segmentgrenzen bei der inhaltsbasierten Segmentierung ist die Wortverteilung in den einzelnen Textblöcken. Es wird davon ausgegangen, dass eine Änderung der Häufigkeitsverteilung von Wörtern einer Änderung des Themas entspricht. Die einzelnen Verfahren unterscheiden sich im Wesentlichen in der genauen Berechnung der Häufigkeitsverteilungen sowie der Methodik zur Bildung der Textblöcke, die zur Analyse herangezogen werden.

Für das von Hearst [66] vorgestellte Verfahren *TextTiling* werden drei wesentliche Schritte durchgeführt:

1. Während der Vorverarbeitung wird der zu segmentierende Text zunächst tokenisiert. Token, die über den gesamten Text mit einer hohen Häufigkeit auftauchen, sogenannte *Stoppwörter*, werden verworfen. Anschließend wird der Text in n Tokensequenzen gleicher Länge unterteilt (im Folgenden $t_1, \ldots, t_n$).

2. Für jedes Paar von adjazenten Tokensequenzen (t_i, t_{i+1}) wird eine Ähnlichkeitsbewertung $w_{i,i+1}$ vorgenommen. Diese Bewertungen werden mittels zwei verschiedener Ansätze bestimmt:

   a) Die Tokensequenzen werden zu Blöcken zusammengefasst. Diese Aggregation zu Blöcken wird unter Verwendung eines fortlaufenden Fensters realisiert. Für jedes Paar von adjazenten Blöcken wird eine Ähnlichkeit dieser Blöcke bestimmt. Die Ähnlichkeit ist hoch, wenn eine große Zahl übereinstimmender Token in beiden Blöcken auftauchen.

   b) Die Anzahl der erstmals im Text auftauchenden Token im aktuell betrachteten Paar von Tokensequenzen ergibt die Bewertung.

3. Zur Identifikation der Segmentgrenzen werden nun die zuvor bestimmten Ähnlichkeitsbewertungen $w_{i,i+1}$ für die Paare von Tokensequenzen verwendet. Für jedes Paar von Tokensequenzen t_i, t_{i+1} wird dessen Bewertung $w_{i,i+1}$ und die Bewertungen $w_{i-1,i}$ und $w_{i+1,i+2}$ der benachbarten Paare betrachtet und der akkumulierte Wert $g_{i,i+1} = (w_{i,i+1} - w_{i-1,i}) + (w_{i+1,i+2} - w_{i,i+1})$ bestimmt. Wenn dieser Wert $g_{i,i+1}$ sehr groß ist, wird zwischen den Tokensequenzen t_i und t_{i+1} eine Segmentgrenze angenommen, da eine große Differenz der Wortverteilungen zwischen den beiden Sequenzen festgestellt wurde. Die finale Segmentgrenze wird am nächstliegenden Textabsatz zum Übergang zwischen Tokensequenz t_i und t_{i+1} gesetzt.

Eine Evaluation unter Verwendendung eines Goldstandards bestehend aus 12 Testtexten zeigte eine Precision zwischen 52 und 71% sowie einen Recall zwischen 59 und 78% bei Verwendung der beiden vorgestellten Alternativen 2.a) und 2.b), unterschiedlichen Parametern für die Anzahl der Tokensequenzen n sowie Variationen des Grenzwerts g ab dem eine Segmentgrenze gesetzt wird. Die besseren Ergebnisse konnten mit Variante 2.a) erzielt werden.

Ein ähnliches Verfahren wurde von Choi [29] vorgestellt. Das vorgestellte Verfahren *C99* weist zunächst Ähnlichkeiten zu Schritt 1 des TextTilling-Verfahrens auf. Die Tokensequenzen entsprechen hier jedoch jeweils genau einem Satz und sind somit von unterschiedlicher Länge. Im Folgenden wird bei C99 ähnlich wie bei Schritt 2.a) des TextTiling-Verfahrens vorgegangen, es erfolgt jedoch keine Aggregation zu Blöcken. Auf Basis der Termfrequenzen wird die Ähnlichkeit adjazenter Sätze bestimmt. Diese absoluten Ähnlichkeitswerte werden anschließend lokal sortiert und somit erhält jedes Satzpaar einen Ähnlichkeitswert in Relation zu anderen Satzpaaren in der textuellen Umgebung. Unter Verwendung der sich daraus ergebenden Ränge als Merkmale wird im Folgenden ein Clustering der Sätze vorgenommen. Jedes Cluster entspricht abschließend einem Textsegment.

Bei Vergleich von C99 mit TextTiling zeigte C99 eine geringere Fehlerrate. Bei Betrachtung der Berechnungskomplexität schnitt TextTiling aufgrund des nicht benötigten Clusterings jedoch besser ab [29].

Riedl und Biemann [133] konnten zeigen, dass TextTiling und C99 bei zusätzlicher Betrachtung der Verteilungen von Themen unter Verwendung eines *Topic Models* [16], wie der *Latent Dirichlet Allocation* [17], statt der ausschließlichen Verwendung der Wortverteilungen bessere Segmentierungsergebnisse aufweisen können. Das Themenmodell muss dafür zunächst durch einen unannotierten Trainingsdatensatz der gleichen Domäne trainiert werden. Der Vorteil einer solchen Anreicherung konnte ebenso von Du et al. [42] gezeigt werden.

In weiterführenden Arbeiten wurde die Verwendung der vorgestellten Textsegmentierungsverfahren in unterschiedlichen Anwendungen untersucht. So wurde die Unterscheidung zwischen lokalen und globalen Themen eines Dokuments umgesetzt [67], die automatisierte Erstellung von Zusammenfassungen von Texten [90], die Erstellung von E-Learning-Kursen [120] sowie die Erkennung von Ereignissen im Bereich der Informationsextraktion [21] vorgeschlagen. Weiterhin wurde eine Abwandlung der Verfahren zur Erkennung von Spam verwendet [163].

Verfahren zur inhaltsbasierten Segmentierung erlauben eine Segmentierung ohne Verwendung von manuell annotierten Trainingsdaten. Allerdings erlauben sie keine Bestimmung des semantischen Typs der einzelnen Segmente. Diese Bestimmung könnte per weitergehender Klassifikation erreicht werden. Die Verfahren basieren jeweils auf der Wort- oder Themenverteilung in den betrachteten Texten. Unterschiede in der Verteilung lassen sich erst bei längeren Segmenten zuverlässig identifizieren, da die erkannten Unterschiede in kürzeren Segmenten auch statistische Ausreißer darstellen könnten. Das Verfahren TextTiling setzt außerdem das Vorhandensein von Absätzen im Text voraus.

3.2 STRUKTURIERUNG VON TEXTEN

Verfahren zur *Strukturierung* von Texten befassen sich damit, wie die in einem Textdokument enthaltenen Informationen in eine strukturierte Form überführt werden können. Insbesondere ist es hier, im Gegensatz zur Segmentierung von Texten, relevant, dass der semantische Typ der Segmente, also die Bedeutung für den Text, in der strukturierten Form beschrieben ist.

Die Strukturierung von Texten kann mit zwei verschiedenen Klassen von Ansätzen durchgeführt werden. Bei Verfahren der ersten Klasse (beschrieben in Abschnitt 3.2.1) liegt vor der Strukturierung eines Dokuments kein Modell vor, anhand dessen die Strukturierung durchgeführt werden soll. Dies bedeutet, dass nicht bekannt ist, welche Informationen nach der Strukturierung in der strukturierten Form vorliegen können. Dies hat einerseits den Vorteil, dass kein Modell manuell definiert werden muss, andererseits jedoch den Nachteil, dass eine Verwendung der strukturierten Daten im Anschluss nicht ohne weiteres möglich ist, da nicht einheitlich definiert ist, welches der resultierenden Segmente welche Information trägt. Für Verfahren der zweiten Klasse (beschrieben in Abschnitt 3.2.2) muss im Gegensatz dazu jedoch ein Modell vorliegen, anhand dessen eine Strukturierung vorgenommen wird. Dieses Modell muss definiert werden, was unter Umständen das Hinzunehmen eines Domänenexperten erforderlich macht, jedoch kann die resultierende strukturierte Form direkt für weitere Anwendungen verwendet werden, da einheitlich bekannt ist, welche Informationen die einzelnen Segmente tragen können.

3.2.1 *Strukturierung ohne Verwendung eines Modells*

Déjean und Meunier [36] schlagen vor, Dokumente anhand ihres Inhaltsverzeichnisses zu strukturieren. Hierzu muss zunächst eine Identifikation des Inhaltsverzeichnisses stattfinden, wozu der Grad der Überlappung von Textfragmenten des potentiellen Inhaltsverzeichnisses mit Textfragmenten des Dokumentes analysiert wird. Bei einer hohen Zahl von Übereinstimmungen wird davon ausgegangen, dass das Inhaltsverzeichnis identifiziert wurde. Die Annahme ist, dass das Inhaltsverzeichnis aus Überschriften der Kapitel besteht, welche die Bedeutung des Kapitels beschreiben. Anschließend wird aus den Elementen des Inhaltsverzeichnisses eine hierarchische Struktur erstellt und eine Zuordnung zu den einzelnen textuellen Segmenten, also den Kapiteln und Abschnitten, vorgenommen. Das Verfahren wurde evaluiert mit Dokumenten, die aus 30 bis 600 Textseiten bestehen, und zeigte hier hervorragende Ergebnisse mit Precision und Recall über 97%. Weiterhin wurde vorgestellt, wie das Verfahren verwendet werden kann, um aus PDF-Dokumenten strukturierte XML-Dokumente zu erzeugen, bei denen der Lesefluß der PDF-Dokumente erhalten bleibt [37].

Eine Strukturierung von Texten unter Verwendung von Schlüsselphrasen und automatisch generierter Inhaltsverzeichnisse wurde von Erbs et al. vorgestellt [44, 162]. In einer initial durchgeführten Nutzerbefragung hat sich der Wunsch der Nutzer nach diesen Strukturierungsmitteln gezeigt, um ein besseres und schnelleres Verständnis von Texten durch den Leser zu ermöglichen. Schlüsselphrasen repräsentieren die für den Text relevanten Konzepte, wohingegend Inhaltsverzeichnisse die sequentielle, inhaltliche Struktur der Texte repräsentieren können. Weiterhin werden

Links eingesetzt, um Phrasen im zu strukturierenden Text mit weiteren Quellen zu verbinden, die es dem Leser ermöglichen, weitergehende Informationen einzuholen.

Eine Strukturierung von Texten ohne Verwendung eines Modells erlaubt im Allgemeinen keinen einheitlichen Zugriff auf spezifische Informationen im Text, da nicht definiert werden kann, welche Typen von Informationen für eine Anwendung der strukturierten Daten relevant sind und nicht zugesichert werden kann, ob die relevanten Informationen entsprechend identifiziert wurden.

3.2.2 *Strukturierung unter Verwendung eines Modells*

Eine Methode zur Strukturierung von Zeitungsannoncen wurde von Embley et al. [43] vorgestellt. Betrachtet werden Zeitungsannoncen mit Kfz-Verkaufsanzeigen und Stellenangeboten. Diese Annoncen zeichnen sich durch ihre Kürze aus, was in einer sehr einheitlichen Vorstrukturierung und beschränkter Terminologie resultiert. Neben den relevanten Informationen sind meist nur wenige weitere Informationen zu finden. Das Verfahren verwendet eine größere Zahl an Listen und Regeln zur Identifikation der relevanten Informationen. In der Evaluation wurde für Kfz-Verkaufsanzeigen eine durchschnittliche Precision von 99% und ein Recall von 94% erreicht. Die Precision bei der Identifikation relevanter Informationen in Stellenanzeigen lag bei 98%, während der Recall bei 84% lag.

Webseiten von Online-Shops haben in der Regel für die einzelnen Produkte eine einheitliche Darstellungsform. Insbesondere sind häufig Sammlungswebseiten zu finden, welche für die gelisteten Produkte die jeweils gleichen Informationen in einheitlicher Struktur darstellen. Abbildung 5 zeigt ein solches Beispiel für den Verkauf von Büchern. Die Webseite ist vorstrukturiert und zeigt für jedes einzelne Produkt relevante Informationen wie Titel, Erscheinungsdatum und Autor in einem einheitlichen Format. Webseiten dieser Art werden meist auf Basis von Datenbanken mit Inhalten befüllt [31]. Diese Datenbanken sind jedoch nicht öffentlich verfügbar. Um Anwendungen, welche auf strukturierte Daten zugreifen, dennoch zu ermöglichen, wird in einigen Arbeiten adressiert, wie der Inhalt dieser Webseiten in strukturierte Form überführt werden kann.

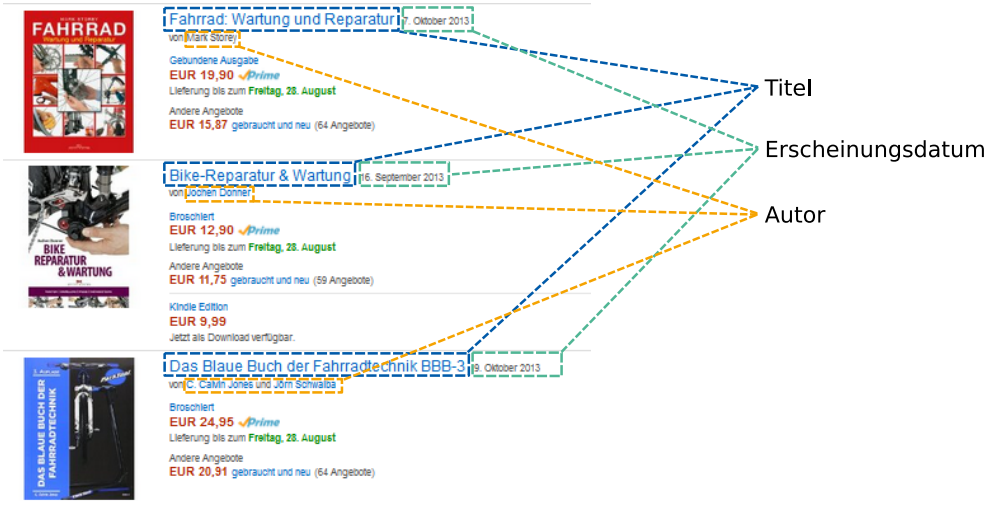


Abbildung 5: Beispiel einer vorstrukturierten Sammlungswebseite, *Quelle: http://www.amazon.de, abgerufen am 27.08.2015*

RoadRunner [31] benötigt als Input mehrere Webseiten, welche gleiche Einträge mit ähnlicher Struktur enthalten. Beispielsweise könnten dies zwei Online-Shops für Bücher sein; beide Shops haben eine ähnliche Menge an Büchern im Angebot, aber diese Menge ist in variierenden Formaten dargestellt. Das Verfahren versucht Übereinstimmungen zwischen den Seiten zu identifizieren. Auf Basis dieser Übereinstimmungen werden typische Muster der DOM-Trees bestimmt, die zur Strukturierung verwendet werden können. Weiterhin werden Optimierungen vorgeschlagen, um das Finden von Übereinstimmungen zu vereinfachen und somit die algorithmische Komplexität zu reduzieren.

Álvarez et al. [4] stellen einen Ansatz vor, bei dem vorab das Muster der konkreten Webseite, die strukturiert werden soll, bekannt sein muss. Dies könnte gegeben werden, indem auf der Webseite einzelne Einträge manuell mit semantischen Informationen annotiert werden. Aus dieser Annotation kann dann die Baumstruktur des DOM der spezifischen Seite gebildet und als Muster zur Extraktion verwendet werden. Die zu strukturierende Webseite wird dann auf das Vorkommen dieses Musters geprüft. Für alle potentiellen Übereinstimmungen wird eine Bewertung auf Basis der Ähnlichkeit zum Muster gegeben und die Übereinstimmung mit der höchsten Bewertung wird als Treffer angenommen. Da Webseiten häufig auf einer einzelnen Seite mehrere Datensätze des gleichen Typs beschreiben, wird das Verfahren zur Bestimmung der Musterübereinstimmung rekursiv durchgeführt. Das Verfahren erreichte in der Evaluation eine Precision von ~98% und einen Recall von ~98%. Unter gleichen Testbedingungen erreichte das zuvor vorgestellte System RoadRunner ~87% Precision und ~66% Recall.

Im Gegensatz zu RoadRunner und dem von Álvarez et al. vorgeschlagenen Verfahren verwendet das von Zheng et al. [179] vorgeschlagene Verfahren vorrangig den Inhalt der zu strukturierenden Elemente und nicht die sie umgebenden Markup-Elemente. Die Elemente mit der höchsten Entropie auf einer Webseite werden dabei als die zu strukturierenden Elemente angenommen, da Elemente mit niedriger Entropie als wiederkehrende beschreibende Elemente angenommen werden. Die Semantifizierung dieser Elemente, also die Zuordnung zum Typ der Information, die sie enthalten, erfolgt im Anschluss. Dafür müssen für jeden dieser Typen einige Beispielelemente gegeben werden. In der genannten Arbeit [179] findet sich eine Übersicht weiterer Systeme zur Strukturierung der beschriebenen Sammlungswebseiten, die unter Verwendung einer Datenbank mit Inhalten befüllt wurden.

Die vorgestellten Verfahren zur Strukturierung von Texten unter Verwendung eines vordefinierten Modells der enthaltenen Informationen gehen von einer Vorstrukturierung der Dokumente aus und erlauben somit nur eine geringe Heterogenität der zu strukturierenden Dokumente. Weiterhin wird die Adaptivität der Verfahren nicht als Ziel betrachtet, eine Anwendung auf Dokumente aus einer neuen Quelle oder einer neuen Domäne ist nur mit wesentlichen Änderungen möglich.

3.3 INFORMATIONSEXTRAKTION

Informationsextraktion befasst sich mit der Extraktion von Informationen aus natürlichsprachlichem Text. Die extrahierten Informationen können im Folgenden für weitergehende Anwendungen verwendet werden. Jurafsky und Martin [75, Seite 759] definieren Informationsextraktion wie folgt:

„[...] a series of techniques that extract limited kinds of semantic content from text."[3]

Grishman [58, Seite 1] weist weitergehend explizit darauf hin, dass dies nicht gleichzusetzen ist mit einem allgemeinen, maschinellen Verständnis der Texte:

„Information extraction is a more limited task than 'full text understanding'. In full text understanding, we aspire to represent in an explicit fashion all the information in a text. In contrast, in information extraction we delimit in advance, as part of the specification of the task, the semantic range of the output: the relations we will represent, and the allowable fillers in each slot of a relation."[4]

Das breite Forschungsthema der Informationsextraktion besteht aus zahlreichen Teilgebieten, welche sich jeweils auf eine spezielle Technik zur Extraktion oder auf spezifische zu extrahierende Informationen fokussieren. Diese Gebiete sind nicht disjunkt und somit schwer voneinander abzugrenzen. Klassischerweise werden in der Informationsextraktion unäre Relationen betrachtet, bei denen aus einem Text alle Token oder Tokensequenzen `t` extrahiert werden sollen, für die eine Relation `istVomTypX(t)` zu `True` ausgewertet wird. Der Typ `X` kann sowohl relativ generisch, wie beispielsweise „Stadt", „Person" oder „Datum", oder auch spezifischer, wie „Stadt in Deutschland", „SPD-Politiker" oder „Geburtsdatum", sein. Allgemeiner ist der Forschungsbereich der Relationsextraktion, bei der n-äre Relationen mit $n \geqslant 2$ betrachtet werden [143]. Vorstellbar sind beispielsweise binäre Relationen wie `istStadtIn(`t_1`,`t_2`)` oder `istPreisVon(`t_1`,`t_2`)`. Ziel ist es, Tokensequenzen t_1 und t_2 zu finden unter denen die Relationen zu `True` ausgewertet werden.

Umfassende Übersichten über Arbeiten im Bereich der Informationsextraktion wurden von Sarawagi [143], Piskorski und Yangarber [125] sowie Small und Mesker [156] erstellt. Arbeiten zur Verknüpfung der extrahierten Informationen mit Wissensbasen wurden von Gangemi [55] zusammengefasst. Im Folgenden wird auf drei Bereiche der Informationsextraktion eingegangen, welche für die vorliegende Arbeit aufgrund ihrer inhaltlichen Nähe relevant sind. *Eigennamenerkennung* (beschrieben in Abschnitt 3.3.1) befasst sich mit der Identifikation von Eigennamen. Diese tauchen in den im Rahmen dieser Arbeit adressierten textuellen Dokumenten auf. In der *ontologiebasierten Informationsextraktion* (beschrieben in Abschnitt 3.3.2), dem zweiten betrachteten Bereich, werden Ontologien verwendet, um sowohl relevante Informationen zu extrahieren als auch um die extrahierten Informationen zu speichern [173]. Als strukturierte Darstellung der in dieser Arbeit adressierten Dokumente könnten Ontologien verwendet werden. Weiterhin bieten sich Ontologien zur Verwendung in der eigentlichen Extraktion an, da sie in vielen Domänen vorliegen und somit eine Adaptivität der Verfahren durch Austausch der Ontologie ermöglicht werden könnte. Der dritte betrachtete Bereich, die *Offene Informationsextraktion* (beschrieben in Abschnitt 3.3.3), weicht die von Grisham [58] definierte Anforderung an Informationsextraktionssysteme, dass bereits vor der Extraktion bekannt ist, welche Informationen extrahiert werden, auf. Es können auch Informationen extrahiert werden,

3 *„Eine Menge von Techniken, die eine begrenzte Anzahl an Arten semantischen Inhalts aus Texten extrahieren."* (freie Übersetzung durch den Autor)

4 *„Informationsextraktion ist eine enger umfasste Aufgabe als das volle Verständnis eines Textes. Für das volle Verständnis eines Textes wird es angestrebt, jegliche Information im Text explizit zu repräsentieren. Im Gegensatz dazu, wird bei der Informationsextraktion im vorhinein die Menge möglicher zu extrahierender semantischer Werte eingeschränkt. Dies betrifft die Relationen, die repräsentiert werden sollen, und die möglichen Argumente der Relationen."* (freie Übersetzung durch den Autor)

deren Typ nicht vorab definiert wurde. Dies erlaubt eine Extraktion in neuen Domänen und somit eine gute Domänenadapativität.

3.3.1 *Erkennung von Eigennamen*

Eigennamenerkennung[5] hat sich als eines der ersten eigenständigen Forschungsthemen im Bereich der Informationsextraktion etabliert [113]. Auch in aktuellen Systemen zur Informationsextraktion wird häufig zunächst eine Erkennung von Eigennamen durchgeführt [75]. In frühen Arbeiten zum Thema wurden vorrangig manuell erstellte Regeln verwendet, um Eigennamen zu identifizieren. Aufgrund des hohen manuellen Aufwands für die Erstellung der Regeln hat sich im weiteren Verlauf die Verwendung von Verfahren des maschinellen Lernens etabliert [113].

Häufig werden sequentielle Modelle, wie *Conditional Random Fields (CRF)* oder *Hidden Markov Models (HMM)*, als Repräsentation des Problems der Eigennamenerkennung verwendet [130]. Für eine Tokensequenz $x = (x_1, \ldots, x_N)$, die als Eingabe agiert und dem Ausgangstext entspricht, ist eine Ausgabesequenz $y = (y_1, \ldots, y_N)$ zu finden. Die Elemente y_i sind die Vorhersagen des Tokentyps, also der Klasse, der einzelnen Token. Im Falle der Eigennamenerkennung sind typischerweise *Person*, *Stadt* oder *Produkt* valide Typen. Zur Identifikation der Ausgabesequenz können sowohl Informationen über die einzelnen Token, ihrer benachbarten Token, der bisher zugeordneten Tokentypen als auch sonstiger Textelemente oder externe Informationen mit berücksichtigt werden. Ratinov und Roth [130] erläutern, dass es beim Design von Methoden zur automatischen Erkennung von Eigennamen vier wesentliche Designentscheidungen zu treffen gibt. Diese betreffen

1. die Repräsentationen x_i der Token,
2. das Verfahren zum Lösen des durch das Modell definierten Problems (die Zuordnung von Token zu Typen von Eigennamen),
3. das Einbringen von Informationen, die nicht in der direkten textuellen Nähe der einzelnen Token stehen, und
4. das Einbringen von externen Wissensquellen.

Das populäre System *KnowItAll* zur Eigennamenerkennung wurde von Etzioni et al. [45] vorgestellt. Das Verfahren baut auf den von Hearst [65] vorgeschlagenen Extraktionsmustern zur Erkennung von Hyponymien auf. Typische solcher Extraktionsmuster sind „`<NP1> is a <NP2>`“ oder „`<NP1> and other <NP2>`“, wobei `NP` eine Nominalphrase bezeichnet. Aus einem Satz „A dog is a mammal.“ lässt sich mittels des ersten Musters die Relation `isA(dog, mammal)` identifizieren. Im System KnowItAll werden solche Extraktionsmuster verwendet. Weiterhin kommt ontologisches Wissen über Entitäten mit Eigennamen zum Einsatz. Die extrahierten Informationen werden unter Verwendung großer Mengen von Webseiten, auf die mittels einer Suchmaschine zugegriffen wird, validiert. Die Kalibrierung dieses Validierungsschritts erfolgt mittels annotierter Trainingsdaten, somit gehört KnowItAll zur Klasse der überwachten Lernverfahren. In Folgearbeiten wurde eine unüberwachte Version von KnowItAll vorgestellt [46].

5 engl. *Named Entity Recognition*

Sowohl im Bereich der Eigennamenerkennung als auch bei sonstigen Aufgaben zur Verarbeitung natürlichsprachlicher Texte liegt der Fokus meist auf englischsprachigen Texten. Nothman et al. [119] zeigen, wie Wikipedia aufgrund seiner guten Abdeckung in zahlreichen Sprachen zur Erkennung von Eigennamen auch in nicht-englischen Texten verwendet werden kann. In diesem multilingualen Ansatz werden Wikipedia-Seiten, die ein Konzept mit Eigennamen beschreiben, als Trainingsdaten für ein maschinelles Lernverfahren verwendet. Neben textuellen Merkmalen werden auch Linkstrukturen innerhalb der Wikipedia berücksichtigt. In der Evaluation ist zu erkennen, dass die Güte des Verfahrens stark von der betrachteten Sprache abhängt. Während die besten Ergebnisse in englischsprachigen Texten (F1-Maß: ~85%) erzielt werden, werden die schlechtesten Ergebnisse in deutschsprachigen Texten erzielt (F1-Maß: ~67%); die Ergebnisse für Spanisch, Niederländisch und Russisch liegen mit ~79%, ~78% und ~80% im Mittelfeld. Zahlreiche weitere Arbeiten fokussieren sich auf die Erkennung von Eigennamen in einzelnen Sprachen, wie beispielsweise Deutsch [13], Türkisch [166], Arabisch [155], Hebräisch [111].

Aktuelle Arbeiten beschäftigen sich häufig mit der Identifikation von Eigennamen in den maximal 140 Zeichen langen Tweets des sozialen Netzwerks Twitter [87, 92, 134].

Eigennamen sind häufig ambigue, das heißt für eine Tokensequenz, die einen Eigennamen beschreibt, ist nicht zwangsläufig ersichtlich, auf welche Entität diese referenziert. Beispielsweise ist „Neustadt“ ambigue, da es mehrere Städte mit diesem Namen gibt, auch Personennamen sind häufig nicht eindeutig. Im Satz „Bundespräsident Köhler besuchte New York“ lässt sich für einen menschlichen Leser eindeutig identifizieren, um welche Person es sich handelt. Das von einem System zur Eigennamenerkennung identifizierte Token „Köhler“ ist jedoch nicht eindeutig. Zur Auflösung dieser Uneindeutigkeit können Verfahren der *Named Entity Disambiguation* oder der verallgemeinerten *Word Sense Disambiguation* [114] genutzt werden. Unter Verwendung des textuellen Kontexts ermöglichen sie die Auflösung der Uneindeutigkeit. Hierzu wird häufig enzyklopädisches Wissen aus Wissensbasen wie Wikipedia [33, 64] oder DBpedia [34, 69] verwendet.

3.3.2 *Ontologiebasierte Informationsextraktion*

Ontologiebasierte Informationsextraktion[6] beschäftigt sich damit, wie Ontologien zur Extraktion genutzt werden können oder wie aus Texten extrahierte Informationen in einer Ontologie gespeichert werden können [173].

Ontologien sind vielfältig einsetzbare Wissensrepräsentationen bestehend aus Knoten, welche Konzepte repräsentieren, und Relationen zwischen den Knoten, die die Beziehung zwischen den Konzepten beschreiben. Die manuelle Erstellung von Ontologien erfordert umfassendes Wissen der jeweils zu repräsentierenden Domäne und der Techniken zur Formalisierung von Konzepten und deren Zusammenhänge [32]. Um den Aufwand der Erstellung zu reduzieren, ist eine automatisierte Erstellung von Ontologien oder eine automatische Unterstützung bei der manuellen Erstellung von hoher Relevanz.

Faatz et al. [49] stellen ein frühes Verfahren zur Anreicherung von Ontologien mit neuen Begriffen unter Verwendung von Webseiten vor. Neue Begriffe werden auf

6 engl. *Ontology-based Information Extraction*

Basis von *Kollokationen*, also dem gemeinsamen Auftreten von Begriffen mit anderen Begriffen, die bereits in der Ontologie vorhanden sind, hinzugefügt. Hierzu werden lexikalische Merkmale, wie maximale Tokenabstände im Text zwischen den beiden Begriffen berücksichtigt. Unter Verwendung des vorgestellten Verfahrens lassen sich verwandte Begriffe in die Ontologie einfügen, jedoch lässt sich nicht ableiten, in welcher semantischen Relation der neu hinzugefügte Begriff zu den existierenden Begriffen steht. Zahlreiche Arbeiten fokussieren sich auf die Erkennung spezifischer Relationen zwischen den durch Begriffe beschriebenen Entitäten [164]. Die sicherlich am häufigsten betrachteten Relationen sind die *Hyponym-* beziehungsweise *Hypernym*-Relation, welche eine taxonomische Beziehung zwischen Begriffen beschreibt [39, 40, 65, 104], sowie die *Synonym*-Relation, welche die gleiche Bedeutung zweier Begriffe beschreibt [126, 170]. Unter Verwendung der Wissensbasis *Wiktionary* lassen sich multilinguale Ontologien erstellen [108]. Eine Übersicht von Methoden zur automatisierten Erstellung von Ontologien, die neben Wissensbasen häufig unter Verwendung von unstrukturierten Textsammlungen vorgenommen wird, wurde von Cristani et al. [32] vorgestellt. Die große Zahl der Freiheitsgrade bei der Erstellung und Anreicherung von Ontologien beziehungsweise der Speicherung der Ergebnisse der Informationsextraktion machen die Evaluation der Ansätze zu einem nicht-trivialen Problem [102].

Ontologien ermöglichen es, menschliches Wissen in maschinenlesbarer Form zu repräsentieren. Da Texte im Normalfall vom Menschen manuell generiert wurden, ist menschliches Wissen notwendig, um die einzelnen Elemente der Texte interpretieren zu können. Auch um relevante Informationen zu extrahieren ist ein gewisses Maß an Interpretation notwendig. Ontologien können an dieser Stelle zur Unterstützung der Informationsextraktionsverfahren verwendet werden, da aus ihnen das benötigte Wissen entnommen werden kann. Im Folgenden wird auf Verfahren eingegangen, welche ontologisches Wissen zur Extraktion verwenden.

Unterschiedliche Ontologien wurden als Hintergrundwissen zur Extraktion von Informationen verwendet. *WordNet* [109] ist eine lexikalische Ontologie und wurde manuell von Linguisten erstellt. Die daraus resultierende hohe Datenqualität hat zur Verwendung in unterschiedlichen Ansätzen der Informationsextraktion geführt [28, 70]. *Freebase* [20] hat eine hohe Datenabdeckung im Bereich der aktuellen Populärkultur und kann daher gut für moderne Texte verwendet werden [178]. Wimalasuriya et al. [171, 172] schlagen ein Konzept vor, um mehrere Ontologien gemeinsam zur Extraktion von Informationen zu verwenden. Dabei können die einzelnen Ontologien aus einer Domäne unterschiedliche Detailgrade aufweisen oder auch unterschiedliche Informationen über eine Domäne aufweisen. Eine wesentliche Herausforderung hierbei ist das Matching zwischen den einzelnen Ontologien [48], da identifiziert werden muss, welche Knoten verschiedener Ontologien das gleiche Konzept beschreiben. Mit dem Begriff *Linked Data* wird die Verwendung einheitlicher Schemata bei der Verbreitung von strukturierten Daten beschrieben. Die resultierende Datenmenge ist untereinander stark vernetzt und es lassen sich ontologische Strukturen finden. Auch eine solche Form der vernetzten Repräsentation von Wissen kann zur Informationsextraktion verwendet werden [106]. Teilweise wird bei Methoden der ontologiebasierten Informationsextraktion nicht auf existierende Ontologien zurückgegriffen, sondern es werden manuell Ontologien für das spezifische Anwendungsgebiet erstellt und anschließend verwendet [9, 76, 144]. Wenn möglich sollte davon abgesehen werden, da für das Erstellen einer Ontologie das Wissen

eines Domänenexperten notwendig ist. Im Ansatz *Textpresso* [112] wurde eine kleine Ontologie manuell aufgestellt und diese mit vorhandenen Wissensbasen manuell verbunden, um eine Eigennamenerkennung zu realisieren.

Zusammenfassend lässt sich feststellen, dass ontologiebasierte Informationsextraktion bei Verwendung domänenübergreifender Ontologien, wie WordNet, die Extraktion von Informationen unabhängig von der konkreten Domäne ermöglicht. Durch die Verwendung ontologischer Daten, die menschliches Wissen repräsentieren, sind die Verfahren rein statistischen Ansätzen häufig überlegen.

3.3.3 *Offene Informationsextraktion*

Laut Grisham [58] muss bei einem Informationsextraktionssystem bereits in der Spezifikation definiert sein, welche Typen von Informationen extrahiert werden sollen. Der Bereich der *Offenen Informationsextraktion*[7] umfasst Ansätze, bei denen diese Anforderung nicht erfüllt ist. Dieser Forschungsbereich wurde erstmals von Banko et al. [8] beschrieben. Die Verfahren sollen ermöglichen, dass auch Typen von Informationen extrahiert werden können, die nicht vorab explizit definiert wurden. Auf diesem Weg können die Verfahren auch ohne großen manuellen Aufwand zur Extraktion von Informationen in unbekannten Domänen angewandt werden. Die Verfahren ermöglichen insbesondere nicht nur eine Identifikation der Elemente der Relation, sondern auch der Relation selbst. So kann ein System beispielsweise unter Kenntnis der Trainingsdaten (`„Thomas Müller“, „ist Fußballspieler für“, „Bayern München“`)[8] und (`„Marco Sailer“,„ist Fußballspieler für“, „SV Darmstadt 98“`) die Information (`„Dirk Nowitzki“, „ist Basketballspieler für“ „Dallas Mavericks“`) aus Texten, die diese Informationen enthalten, ableiten, auch wenn die Relation `„ist Basketballspieler für“` zuvor unbekannt war.

Banko et al. [8] stellen das Verfahren *TextRunner* vor. Für diesen Ansatz wird zunächst ein linguistischer Parser verwendet, um Trainingsdaten aus Texten zu generieren. Unter Verwendung dieser Trainingsdaten können Token in den Texten als Kandidaten für zu extrahierende Elemente und Relationennamen markiert werden. Diese Markierung erfolgt in gewichteter Form, so dass abschließend nur die Relationen, die in hoher Zahl und mit hoher Wahrscheinlichkeit markiert wurden, als tatsächlich korrekte Relationen angenommen werden. In einer Evaluation wurde TextRunner mit dem zuvor beschriebenen geschlossenen Informationsextraktionssystem KnowItAll (siehe Abschnitt 3.3.1) verglichen. Da KnowItAll eine vorherige Definition der zu extrahierenden Relationen benötigt, wurde die Evaluation auf zehn spezifische Relationen beschränkt. Die Evaluation konnte zeigen, dass trotz der höheren Adaptivität TextRunner eine geringere Fehlerquote als KnowItAll aufweist (12% versus 18%).

ReVerb [47] zielt auf eine geringere Fehlerquote bei der Erkennung ab. Insbesondere soll verhindert werden, dass irrelevante Informationen extrahiert werden. Dazu werden Muster von POS-Tags verwendet und nur extrahierte Informationen, welche häufig in den betrachteten Texten auftauchen, als wahr angenommen. Bei vergleichenden Evaluationen mit TextRunner konnte die Überlegenheit von ReVerb gezeigt werden. Die qualitative Analyse der inkorrekt extrahierten Informationen zeig-

7 engl. *Open Information Extraction*

8 Zur besseren Lesbarkeit wird eine Infix-Notation verwendet, die Darstellung (t_1, r, t_2) beschreibt dabei die Relation $r(t_1, t_2)$.

te, dass häufig die korrekte Relation extrahiert wird, jedoch deren Argumente, also die eigentlich zu extrahierenden Informationen, inkorrekt sind. Das weiterhin in der Publikation vorgestellte Konzept *ArgLearner* verwendet sequenzielle Lernverfahren und Muster von komplexen Satzstrukturen, um die Problematiken bei der Extraktion der Relationsargumente zu adressieren.

Die Wissensbasis *Wikipedia* wird im vom Wu und Weld [177] vorgestellten Verfahren zur offenen Informationsextraktion verwendet. Hierzu werden Infoboxen der Wikipedia genutzt. Infoboxen enthalten in strukturierter Form Informationen zu verschiedenen Entitäten. Infoboxen für Artikel über Städte enthalten Informationen wie die Anzahl der Einwohner, den Name des Bürgermeisters oder das Land, in dem sich diese Stadt befindet. Da diese Informationen jeweils meist auch im unstrukturierten Text eines Wikipedia-Artikels zu finden sind, können diese Paare aus jeweils einem Element in der Infobox und einem Beispiel aus dem Fließtext als Trainingsdaten für ein System zur Relationserkennung verwendet werden. Unter Verwendung der POS-Tags und der Satzstrukturen wird das System trainiert, um relationsunabhängig Informationen zu extrahieren. Das Verfahren ist beschränkt auf die Relationen, die in den Wikipedia-Infoboxen zulässig sind. Deren Anzahl ist jedoch sehr groß [51] und die Menge der zulässigen Relationen ist erweiterbar, weshalb von einem offenen Informationsextraktionssystem gesprochen werden kann.

Die Verfahren der Offenen Informationsextraktion zeichnen sich zwar durch eine hohe Domänenadaptivität aus, da sie auch bisher unbekannte Typen von Informationen extrahieren können, jedoch lässt sich nicht definieren, welche Informationen extrahiert werden sollen. Dies führt dazu, dass Informationen, die für eine konkrete Anwendung relevant sind, mitunter nicht erkannt werden.

3.4 DOMÄNENADAPTIVE KLASSIFIKATION VON TEXTEN

Verfahren zur Klassifikation von Texten oder textuellen Segmenten benötigen meist annotierte Trainingsdaten (vergleiche Abschnitt 2.2.2). Die Erstellung solcher annotierten Trainingsdaten ist aufgrund des hohen manuellen Aufwands der Annotation teuer und gleichzeitig fehleranfällig [71]. Aus diesem Grund sollte die Menge der zu annotierenden Texte möglichst klein gehalten werden.

Verfahren der Textklassifikation nutzen die unterschiedlichen Häufigkeitsverteilungen der Wörter einer Sprache in unterschiedlichen Klassen [2]. Das Wissen über die jeweilige Verteilung der Wörter in den spezifischen Klassen wird über die Trainingsdaten erlangt. Die Verteilung der Wörter ist charakteristisch für Klassen und Domänen. Somit kann das Wissen über die Wortverteilungen einer Klasse, das mit Trainingsdaten einer Domäne erlangt wurde, nicht automatisch in einer anderen Domäne als gültig angenommen werden. Da, wie zuvor beschrieben, die Gewinnung dieser Trainingsdaten einen hohen Aufwand benötigt, wurden in der Vergangenheit domänenadaptive Verfahren zur Textklassifikation vorgestellt. Bei diesen Verfahren ist das Ziel, aus dem Wissen über die Verteilung der Worthäufigkeiten in einer Domäne Wissen über die Verteilung von Worthäufigkeiten in einer anderen Domäne abzuleiten.

Zwei häufig genannte Konzepte mit dem Ziel des Übertragens von Wissen aus einer Ausgangsdomäne in eine Zieldomäne sind *Domain Adaptation* und *Transfer Learning*. Patricia und Caputo [124] unterscheiden diese Konzepte wie im Folgenden be-

schrieben. Bei der Domain Adaptation wird die Annahme getroffen, dass die Klassen zwischen Ausgangsdomäne und Zieldomäne identisch sind. Es unterscheiden sich jedoch die Merkmalsverteilungen in den einzelnen Klassen. Im Fall der Textklassifikation bedeutet dies, dass für eine spezifische Klasse unterschiedliche Wortverteilungen in Ausgangsdomäne und Zieldomäne existieren. Dieses Charakteristikum ist auch häufig bei unrepräsentativer Aufteilung eines Datensatzes in Trainings- und Testdaten[9] zu beobachten. Bei Verfahren des Transfer Learnings liegen unterschiedliche Klassen in Ausgangs- und Zieldomäne vor. Die Verteilungen der Wörter, also der Merkmale, sind unterschiedlich, aber ähnlich.

Die Annahme gleicher Zielklassen, die bei Verfahren der Domain Adaptation getroffen wird, schränkt die Anwendungsmöglichkeit der Verfahren in wechselnden Domänen stark ein. Ein häufig betrachtetes Anwendungsgebiet von Verfahren der Domain Adaptation in der Textklassifikation ist die Bestimmung des *Sentiments* [18, 57, 103] eines Texts, also der Haltung eines Autors zum verfassten Text. Die Bestimmung des Sentiments kann domänenunabhängig in die Klassen *positiv, neutral, negativ* vorgenommen werden. Ob die Verfahren tatsächlich einen Mehrwert liefern, ist umstritten [99].

Bei Verfahren des Transfer Learnings, bei denen von unterschiedlichen Klassen in Ausgangs- und Zieldomäne ausgegangen wird [123], können Ähnlichkeiten der Klassen genutzt werden. Rohrbach et al. [135] stellen vor, wie semantische Ähnlichkeitsrelationen beim Transfer von Klassenzugehörigkeiten einer Domäne in eine andere Domäne verwendet werden können. Hierzu werden sowohl Ähnlichkeiten zwischen den Klassen als auch zwischen den zur Klassifikation verwendeten Merkmalen betrachtet. Das Verfahren wurde in unterschiedlichen Varianten zur Klassifikation von Bildern verwendet, kann jedoch auch auf Probleme der Textklassifikation angewendet werden. Insbesondere lässt sich eine solcher Wissenstransfer bei hierarchischen Klassen anwenden [158].

Domänenadaptive Verfahren zur Textklassifikation sind in ihrer Anwendung stark eingeschränkt. Verfahren der Domain Adaptation erfordern zur Übertragung von Wissen aus einer Ausgangsdomäne in eine Zieldomäne homogene, also identische Klassen. Bei der Strukturierung von Texten sind die einzelnen strukturellen Elemente der Texte stark von der jeweiligen Domäne abhängig, so dass diese Annahme nicht vorausgesetzt werden kann. Bei Verfahren des Transfer Learnings müssen wiederum Ähnlichkeiten zwischen den zur Klassifikation verwendeten Merkmalen oder den Merkmalen und den einzelnen Klassen vorliegen, auch dies stellt eine starke Einschränkung dar.

3.5 DISKUSSION UND EINORDNUNG DIESER DISSERTATION

In der vorliegenden Dissertation wird betrachtet, wie heterogene textuelle Dokumente in eine strukturierte Form überführt werden können. Die Struktur ist dabei abhängig von der Domäne, der die Dokumente zugeordnet werden können. Insbesondere soll betrachtet werden, wie Verfahren zur Strukturierung bei möglichst hoher Qualität mit möglichst geringem manuellen Aufwand in einer neuen Domäne

9 In diesem Falle entsprechen die Dokumente im Trainingsdatensatz der Ausgangsdomäne und die Dokumente im Testdatensatz der Zieldomäne.

verwendet werden können. In diesem Kapitel wurde ein Überblick über Arbeiten in Forschungsbereichen mit ähnlichen Zielsetzungen gegeben.

Verfahren der Textsegmentierung (Abschnitt 3.1) erfüllen die Anforderung, dass Dokumente in einzelne strukturelle Elemente zerteilt werden. Eine Semantifizierung der Segmente, also die Zuordnung eines Segments zum darin enthaltenen Typ der Information, könnte im Folgenden mittels Klassifikationsverfahren erfolgen. Layoutbasierte Segmentierungsverfahren (Abschnitt 3.1.1) gehen davon aus, dass die zu strukturierenden Dokumente mittels einer Auszeichnungssprache bereits in vorstrukturierter Form vorliegen und diese Vorstrukturierung gemäß der Semantik der Dokumente vorgenommen wurde. Diese Einschränkungen sind problematisch, da sie zum einen eine Verwendung von Dokumenten, die keine Auszeichnungssprache nutzen, verbieten und zum anderen keine Segmentierung von Fließtextabschnitten erlauben. Diese Einschränkungen gelten für inhaltsbasierte Segmentierungsverfahren nicht (Abschnitt 3.1.2). Deren Nutzbarkeit ist jedoch erst bei längeren Segmenten gegeben, da erst dann statistisch signifikante Unterschiede in den Wortverteilungen der Segmente zu erkennen sind.

Im Gegensatz zur Textsegmentierung wird bei der Textstrukturierung (Abschnitt 3.2) eine Semantifizierung der einzelnen strukturellen Elemente vorgenommen. Bei dem vorgestellten modellfreien Verfahren ist vorab nicht bekannt, welche semantischen Rollen die identifizierten Elemente haben (Abschnitt 3.2.1), was eine einheitliche Nutzung der strukturierten Daten für Anwendungen nur schwer möglich macht. Die Verfahren zielen vorangig auf Nutzung der strukturierten Daten durch den Anwender ab und weniger zur weitergehenden Verwendung in Anwendungen. Bei existierenden modellbasierten Verfahren (Abschnitt 3.2.2) wird ähnlich wie bei layoutbasierten Segmentierungsverfahren von einer Vorstrukturierung der Dokumente ausgegangen. Bei den vorgestellten Strukturierungsverfahren ist diese Vorstrukturierung feingranularer und nicht notwendigerweise durch Verwendung einer Auszeichnungssprache gegeben, sondern kann auch durch eine einheitliche Ordnung des Textes, wie beispielsweise bei Zeitungsannoncen, gegeben sein. Weiterhin muss diese Vorstrukturierung einheitlich für die einzelnen Dokumente einer Domäne sein. Im Rahmen der vorliegenden Arbeit sollen jedoch heterogene Dokumente adressiert werden, welche sich mit den vorgestellten Verfahren nicht strukturieren lassen.

Informationsextraktion erlaubt das Identifizieren relevanter Informationen in Dokumenten (Abschnitt 3.3). Diese entsprechen den zu identifizierenden strukturellen Elementen der Texte. Existierende Verfahren im Bereich Eigennamenerkennung erzielen bereits sehr gute Ergebnisse und sind durch Verwendung enzyklopädischer Daten flexibel einsetzbar (Abschnitt 3.3.1). Ontologiebasierte Verfahren (Abschnitt 3.3.2) können den Extraktionsprozess durch ontologische Daten unterstützen, dies erfordert jedoch die Existenz nutzbarer Ontologien der adressierten Domäne sowie das Filtern der relevanten Daten, die zur Extraktion nutzbar sind. Die Verfahren der offenen Informationsextraktion (Abschnitt 3.3.3) setzen dagegen kein Domänenwissen voraus und können daher in jeglichen Domänen verwendet werden. Es kann jedoch nicht zugesichert werden, dass die für ein Anwendungsszenario relevanten Informationen extrahiert werden. Weiterhin ist zu beachten, dass alle Informationsextraktionsverfahren informationszentrisch vorgehen, so dass also das Ziel die Bestimmung großer Mengen von Informationen aus Dokumentensammlungen ist. Der Ursprung einer einzelnen Information, also das konkrete Dokument, aus dem die Information stammt, ist hierbei irrelevant. Im Rahmen dieser Arbeit soll jedoch ein dokumenten-

zentrischer Ansatz verfolgt werden. Hierbei steht nicht die einzelne Information im Vordergrund, sondern die Strukturierung kompletter Dokumente. Nichtsdestotrotz können Verfahren der Informationsextraktion verwendet werden, um Informationen aus einzelnen Dokumenten zu extrahieren. Hierbei muss jedoch ein hoher Recall gewährleistet werden um dem dokumentenzentrischen Fokus gerecht zu werden und einen hohen Anteil der strukturellen Elemente der einzelnen Dokumente zu erkennen.

Die vorgestellten domänenadaptiven Verfahren (Abschnitt 3.4) zielen auf eine gute Übertragbarkeit in andere Domänen ab. Die einzelnen Verfahren haben jedoch starke Einschränkungen hinsichtlich der Nutzbarkeit, so müssen entweder gleiche Klassen in Ausgangs- und Zieldomäne vorliegen oder Ähnlichkeiten zwischen den Klassen oder den Merkmalen modellierbar sein.

Zusammenfassend ist festzuhalten, dass keine der existierenden Arbeiten in den diskutierten Forschungsbereichen den Zielen der vorliegenden Dissertation, einer domänenadaptiven Strukturierung heterogener textueller Dokumente, gerecht wird. Verfahren der Textsegmentierung ermitteln nicht die semantische Rolle der Information eines Segmentes und erlauben somit kein gezieltes Zugreifen auf spezifische Informationen. Verfahren der Textstrukturierung ermöglichen dies, jedoch nur für sehr spezifische Anwendungsdomänen. Eine Übertragbarkeit in neue Domänen ist daher für diese nicht möglich. Verfahren der Informationsextraktion können verwendet werden, um spezifische Informationen in textuellen Dokumenten, wie beispielsweise Eigennamen, zu erkennen. Geeignet sind die existierenden Verfahren jedoch nicht, um beispielsweise relevante, längere Textsegmente zu identifizieren. Existierende domänenadaptive Verfahren zur Textklassifikation, die zur Identifikation spezifischer Informationen eingesetzt werden könnten, verlangen entweder identische oder zumindest sehr ähnliche Klassen bei einer Domänenadaption, was ihre praktische Nutzbarkeit stark einschränkt.

4

STRUKTURIERUNG TEXTUELLER DOKUMENTE

Wie im vorhergehenden Kapitel dargestellt, existieren zahlreiche Ansätze, welche darauf abzielen, relevante Informationen in textuellen Dokumenten zu identifizieren. Keiner der Ansätze erlaubt jedoch eine domänenadaptive Strukturierung heterogener textueller Dokumente. In diesem Kapitel wird erläutert, wie die im Rahmen dieser Arbeit adressierte umfassend nutzbare Strukturierung aussieht. Dazu wird zunächst ein Modell vorgestellt, welches die Zusammenhänge zwischen den für die Arbeit relevanten Konzepten darstellt. Weiterhin wird auf die adressierten Anwendungsdomänen eingegangen und analysiert, welche Typen von Informationen in wiederkehrender Form in diesen Domänen zu finden sind.

4.1 MODELL UND KONZEPTE

In diesem Abschnitt wird ein Modell vorgestellt, welche als Grundlage für die in dieser Arbeit konzipierten Verfahren zur Strukturierung von textuellen Dokumenten dient und die dafür relevanten Konzepte zueinander in Bezug setzt. Ein Überblick über die verwendeten Konzepte und ihre Zusammenhänge ist Abbildung 6 zu entnehmen. Im Folgenden werden die verwendeten Konzepte und ihre Zusammenhänge erläutert, weiter gefolgt von einem Anwendungsbeispiel zur Verdeutlichung der Konzepte und ihrer Abhängigkeiten (Abschnitt 4.1.1).

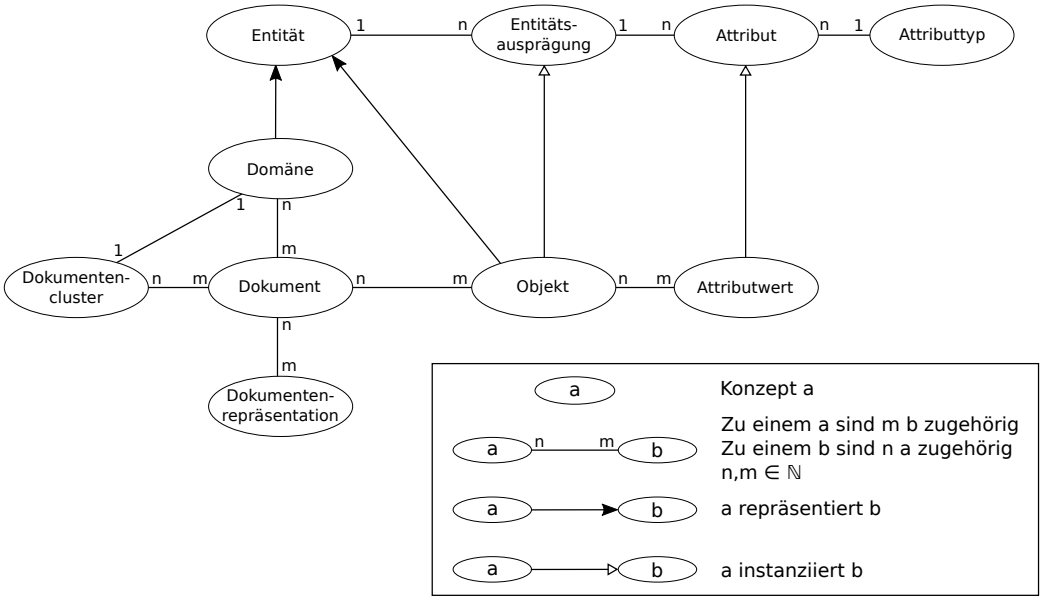


Abbildung 6: Überblick über das zugrunde liegende Modell

Wie in der Motivation dargestellt, soll die Arbeit Grundlagen für verschiedene technische Anwendungen, wie beispielsweise Suchmaschinen oder Empfehlungssysteme, in verschiedenen Domänen zur Verfügung stellen. Der Begriff der *Domäne* be-

zeichnet in diesem Kontext das konkrete Anwendungsgebiet. Der Begriff der *Entität* beschreibt das abstrakte Konzept für die in der Domäne betrachteten Gegenstände[1]. Je nach Nutzungsform beziehungsweise Anwendungsszenario, können unterschiedliche *Entitätsausprägungen* einer Entität relevant sein, die sich hinsichtlich der die Entität beschreibenden Attribute unterscheiden. Objekte sind konkrete Instanziierungen einer Entitätsausprägung. Diese Relation ist analog zur Klassen-Objekt-Relation in der objektorientierten Programmierung zu sehen (vergleiche [54]). Eine Entitätsausprägung ist charakterisiert durch die Menge ihrer *Attribute*, der Eigenschaften, die relevant sind, um Objekte umfassend und geeignet für eine weitere Nutzung zu beschreiben. Jedes Attribut besitzt einen definierten Wertebereich, welcher endlich oder unendlich sowie aufzählbar oder nicht aufzählbar sein kann. Der Wertebereich wird durch den *Attributtyp* eingeschränkt, so kann ein spezifisches Attribut beispielsweise nur aus einem einzelnen Wort oder einem längeren Textsegment bestehen. Die konkreten Merkmalsausprägungen der Objekte werden als *Attributwerte* bezeichnet und lassen sich als beschreibende Zeichenketten repräsentieren, die durch den Attributtyp des Attributs eingeschränkt werden können. Attributwerte lassen sich teilweise durch ihre *interne* und *externe Struktur* beschreiben. Während die interne Struktur das Format des Attributwertes selbst beschreibt, beschreibt die externe Struktur das Format des textuellen Kontextes des Attributwertes im Dokument. So lassen sich beispielsweise E-Mail-Adressen durch ihre sehr regelmäßige interne Struktur beschreiben. Städtenamen lassen sich hingegen auf Basis ihrer externen Struktur beschreiben, da sie häufig auf eine Postleitzahl folgen. Weiterhin können zwei Objekte identische Attributwerte besitzen. Zwei Objekte sind genau dann identisch, wenn sie von gleicher Entitätsausprägung sind und die vollständige Menge ihrer Attributwerte identisch ist. In diesem Fall liegen *Dubletten* vor.

Im Rahmen dieser Arbeit wird betrachtet, wie die Attributwerte eines Objektes aus einem *Dokument* extrahiert werden können. Dabei wird eine n:m Relation zwischen Dokument und Objekt angenommen, das heißt ein Dokument kann mehrere Objekte in textueller Form beschreiben und ein Objekt kann von mehreren Dokumenten zugleich beschrieben werden. Ein Attribut beschreibt die semantische Rolle der im Attributwert definierten Zeichensequenz für das im Dokument beschriebene Objekt. Falls die mit dem Attribut beschriebene Information im Dokument nicht vorhanden ist, so kann ein Attributwert für ein Objekt auch leer sein. Eine im Dokument existierende Zeichensequenz sollte nicht Teil verschiedener Attributwerte eines Objektes sein. Ein Dokument kann mehreren Domänen zuzuordnen sein und kann somit Objekte verschiedener Domänen beschreiben. Dokumente unterliegen keinen äußeren Beschränkungen, weder die Länge noch die Struktur ist vordefiniert. Weiterhin wird angenommen, dass Dokumente in einer Rohtextform vorliegen. Abgesehen von Leerzeichen und Zeilenumbrüchen existieren keine Formatierungen. Da diese Annahme für allgemeine textuelle Dokumente nicht gültig ist, wird der Begriff der *Dokumentenrepräsentation* eingeführt, welcher eine konkrete Datei beschreibt, die das Dokument, also den Rohtext, beinhaltet. Denkbar sind hier unterschiedlichste Dateiformate, wie Rohtexte (.txt), vektorbasierte Seitenbeschreibungen (Portable Document Format (PDF), Postscript (PS) et cetera), anwendungsgebundene Formate (.doc), offene Standards (.docx, .odt et cetera), Auszeichnungssprachen (HTML)

1 Die hier genannten Gegenstände müssen nicht notwendigerweise physischer Natur sein. Ein Beispiel für einen nicht-physischen Gegenstand ist eine Stellenausschreibung.

oder auch Grafikformate (.jpg, .bmp, .png, .jpeg et cetera) mit textuellem Inhalt. Eine Dokumentenrepräsentation beschreibt in der Realität häufig mehrere Objekte einer Entitätsausprägung, wie es bei Sammlungswebseiten der Fall ist, weshalb sich diese in mehrere Dokumente zerteilen lässt. Außerdem kann ein Dokument in mehreren Repräsentationen vorliegen. Daher wird eine n:m Beziehung zwischen Dokument und Dokumentrepräsentation angenommen.

Die Menge aller Dokumente, die Objekte einer gemeinsamen Entität und somit auch einer gemeinsamen Domäne beschreiben, wird als *Dokumentencluster* bezeichnet. Ein Dokumentencluster enthält also Dokumente, die nicht notwendigerweise von einheitlicher äußerlicher Struktur sind, die sich aber anhand der durch die Relation zwischen Entitätsausprägung und Attributen vorgegebenen Struktur strukturieren lassen, so dass die für eine Nutzungsform relevanten Informationen in der strukturierten Repräsentation separiert voneinander in unterschiedlichen Attributen vorliegen.

4.1.1 *Anwendungsbeispiel*

Zum besseren Verständnis des vorgestellten Modells wird in diesem Abschnitt ein Anwendungsbeispiel präsentiert, welches die einzelnen Konzepte und ihre Zusammenhänge verdeutlichen soll.

Betrachtet wird die Domäne *Gebrauchtwagenverkauf*, die Entität ist somit *Gebrauchtwagen*. Verteilt im Internet befinden sich Beschreibungen solcher PKW, beispielsweise in Online-Kleinanzeigenmärkten von Zeitungen, auf dedizierten Webportalen zum Automobilverkauf (beispielsweise mobile.de[2] oder AutoScout24[3]), auf der Webpräsenz von Autohäusern oder auf allgemeinen Kleinanzeigenmärkten (beispielsweise ebay Kleinanzeigen[4]). Die Beschreibungen der Gebrauchtwagen werden in textueller Form auf den Webpräsenzen vorgestellt, häufig sind diese außerdem durch Bilder oder sonstige Medienelemente angereichert. Die einzelnen Webseiten stellen die Dokumentenrepräsentationen dar, wobei auf einer Webseite häufig mehrere Gebrauchtwagen beschrieben werden und somit der textuelle Inhalt mehrere Dokumente umfasst. Weiterhin kann dasselbe Objekt eines Gebrauchtwagens auf mehreren Webseiten beschrieben sein. Relevante Attribute in einer Entitätsausprägung des Gebrauchtwagens könnten beispielsweise *Datum der Erstzulassung*, *Marke*, *Modell*, *Farbe* oder *Kontaktdaten* sein. Tabelle 4 zeigt beispielhafte Attributwerte für zwei unterschiedliche Objekte.

4.2 ANWENDUNGSDOMÄNEN

Als potenzielle Domänen für die im Rahmen dieser Arbeit betrachtete Strukturierung kommen solche Domänen in Frage, die Entitäten und ihre Dokumente betrachten, die sich unter Verwendung ihrer beschreibenden Attribute strukturieren lassen. Zu beachten ist die Sonderstellung zweier Typen von Domänen:

1. Nicht adressiert werden Domänen, in denen zwar textuelle Dokumente im Vordergrund stehen, diese sich aber nicht anhand einer einheitlichen Struktur strukturieren lassen. So lassen sich beispielsweise bei Nachrichtenartikeln

2 http://www.mobile.de/, letzter Zugriff am 24.09.2015
4 http://kleinanzeigen.ebay.de, letzter Zugriff am 24.09.2015

Tabelle 4: Beispielhafte Attributwerte für zwei Objekte einer Entitätsausprägung der Entität *Gebrauchtwagen*

Attribut	Objekt 1	Objekt 2
Marke	VW	Fiat
Modell	Golf	Panda
Farbe	schwarz	rot
Laufleistung	12.000 km	123.000 km
Erstzulassung	12.07.2012	26. März 2003
Preis	14340 €	1.000 Euro
Kontaktdaten	max@mustermann.de	Meyer, 0123-12345
Ausstattung	ASR, ESP, Sitzheizung, Navigationssystem, Schiebedach	ABS

im Allgemeinen nebem dem Titel, dem Thema und gegebenenfalls dem Verfasser keine einheitlichen strukturellen Elemente finden, die sinnvoll in Anwendungen genutzt werden können. Zwar lassen sich gewisse Informationen wie Namen von Personen oder Ortsbezeichnungen häufig finden, aber diese haben keine einheitliche semantische Rolle in den Dokumenten, somit ist keine einheitliche Strukturierung unter Nutzung dieser Informationen möglich.

2. Theoretisch vorstellbar, aber nicht sinnvoll eingesetzt, wäre die Verwendung in einer Domäne mit homogenem Dokumentenformat, in dem alle Dokumente bereits einer einheitlichen Strukturierung unterliegen, wie dies häufig innerhalb spezifischer Plattformen der Fall ist. So werden beispielsweise auf der Verkaufsplattform Amazon[5] Bücher oder sonstige Medien in einem einheitlichen Layout beschrieben. Elemente wie ISBN, Titel, Erscheinungsjahr oder Verlag sind hierbei in jedem Dokument an gleicher Stelle mit gleichem textuellen Kontext zu finden. In einer solchen Domäne liefern einfache regelbasierte Verfahren zur Strukturierung bereits sehr gute Ergebnisse und sind aufgrund der starken Homogenität einfacher zu realisieren als domänenadaptive Verfahren. Dies trifft insbesondere auch für Domänen zu, in denen die strukturellen Elemente bereits als explizite Metadaten ausgezeichnet sind. In der Lernplattform CROKODIL [5] stehen beispielsweise textuelle Dokumente als Lernressourcen im Fokus, diese sind angereichert durch Metadaten wie Autoren oder thematische Tags. Diese Anreicherung durch Metadaten stellt bereits eine einheitliche Struktur dar, welche sich im Datenformat widerspiegelt.

Zusammenfassend lässt sich feststellen, dass ein heterogenes Dokumentenlayout bei homogener Relation zwischen Entitätsausprägungen und Attributen über die Dokumente des betrachteten Dokumentenclusters vorausgesetzt wird.

Um die Nutzbarkeit der im Rahmen dieser Dissertation entwickelten Verfahren in solchen Domänen zeigen zu können, wurde eine Auswahl von fünf Domänen getroffen, die im Folgenden beschrieben werden. Für die einzelnen Domänen werden weiterhin Herausforderungen für die Strukturierung sowie beispielhafte Nut-

5 http://www.amazon.de, letzter Zugriff am 24.09.2015

zungsformen für strukturierte Repräsentationen beschrieben. Die Domänen wurden so ausgewählt, dass sie untereinander möglichst heterogen sind und somit ein breites Spektrum an Anwendungsgebieten aufspannen. Die Dokumentenersteller und -anbieter sind in unterschiedlichen gesellschaftlichen Bereichen zu finden, sowohl in Unternehmen als auch in der Wissenschaft und Bildung. Die Dokumente sind sowohl heterogen hinsichtlich ihres Umfangs als auch ihres Layouts. So lassen sich sowohl Fließtexte bestehend aus kompletten Sätzen als auch vorstrukturierte, stichwortartige Auflistungen finden. Auch hinsichtlich ihrer Formalität unterscheiden sich die Dokumente.

4.2.1 *Stellenanzeigen*

Bei der Suche nach einer neuen Anstellung sind Arbeitnehmer auf Stellenmärkte angewiesen. Stellenanzeigen werden auf unterschiedlichsten Wegen veröffentlicht. Während in der Vergangenheit häufig Stellenangebote über Printmedien, insbesondere Wochen- und Tageszeitungen, verbreitet wurden, veröffentlichen diese Zeitungen heute Stellenanzeigen zusätzlich oder ausschließlich auf ihrer Webpräsenz[6]. Unternehmen[7] und Institutionen[8] bieten häufig auf ihrer Webpräsenz eigene Karriereportale an, auf denen Angebote für offene interne Stellen veröffentlicht werden. Weiterhin gibt es allgemeine Karriereportale, sowohl von der Bundesagentur für Arbeit[9] als auch von kommerziellen Anbietern[10] auf denen Stellenangebote veröffentlicht werden.

Hinsichtlich ihres Layouts weisen die veröffentlichten Dokumente eine große Heterogenität auf. Einige der Portale nutzen standardisierte Layouts, insbesondere auf den allgemeinen Karriereportalen wird häufig eine tabellarische Form verwendet, in der einzelne Informationen explizit benannt werden. Der Grad der Vorstrukturierung variiert jedoch stark und über Portale hinweg wird kein einheitliches Layout verwendet. Innerhalb interner Stellenportale ist häufig ein einheitliches Layout zu finden, wobei die große Zahl der Unternehmensportale zu einer hohen Heterogenität führt. Auf Zeitungsportalen sind sowohl Veröffentlichungen im Layout der Unternehmen als auch einheitlich formatierte Stellenangebote zu finden.

Tabelle 5 gibt einen Überblick über wiederkehrende Attribute in der Domäne *Stellenanzeigen*.

Bei der Suche nach einer möglichen Arbeitsstelle ist der Arbeitnehmer darauf angewiesen, die unterschiedlichen Portale nacheinander zu durchsuchen. Hierbei ist der Suchende mit mehreren Herausforderungen konfrontiert. Die Vielzahl an Arbeitgebern macht es praktisch unmöglich, die Portale der einzelnen Arbeitgeber zu durchsuchen, obwohl auf diesen Portalen die größte Wahrscheinlichkeit für die Veröffentlichung von Stellen des jeweiligen Arbeitgebers ist. Weiterhin ist es ohne ausführliches Studium der ausgeschriebenen Stellen für den Suchenden nicht offensichtlich, wel-

6 beispielsweise http://jobs.zeit.de, http://fazjob.net oder http://jobs.echo-online.de/, letzter Zugriff jeweils am 24.09.2015

7 beispielsweise http://www.merck.de/de/karriere/karriere.html, http://your.bosch-career.com oder https://jobs.softwareag.com/, letzter Zugriff jeweils am 24.09.2015

8 beispielsweise http://www.klinikumfrankfurt.de/service/stellenangebote.html oder http://www.intern.tu-darmstadt.de/dez_vii/stellen/, letzter Zugriff jeweils am 24.09.2015

9 https://jobboerse.arbeitsagentur.de, letzter Zugriff am 24.09.2015

10 beispielsweise https://www.staufenbiel.de, https://www.interamt.de, letzter Zugriff jeweils am 24.09.2015

Tabelle 5: Relevante Attribute in der Domäne *Stellenanzeigen*

Attribut	Beschreibung
Titel	Bezeichnung der zu vergebenden Stelle
Startdatum	gewünschtes Eintrittsdatum
Aufgaben	Beschreibung der zu erfüllenden Aufgaben
Anforderungen	Anforderungen an den Bewerber
Abschluss	erforderlicher Abschluss
Vergütung	Vergütung für die ausgeschriebene Stelle
Arbeitgeber	Name des Arbeitgebers
Arbeitgeberbeschreibung	Beschreibung des Arbeitgebers
Einsatzort	Standort des Unternehmens für den die Stelle ausgeschrieben ist
Berufsfeld	thematische Kategorisierung der Stelle
Ansprechpartner	Name einer Kontaktperson beim Arbeitgeber
E-Mail-Adresse	E-Mail-Adresse des Ansprechpartners
Telefonnummer	Telefonnummer des Ansprechpartners
Anschrift	postalische Anschrift des Ansprechpartners

che Arbeitgeber für ihn geeignete Stellen anbieten. Die Bedeutung einer Zeichensequenz kann in Abhängigkeit des zuzuordnenden Attributs variieren, beispielsweise können Städtenamen in der Arbeitgeberbeschreibung auftauchen, die aber nicht für den Einsatzort relevant sind. Dies erfordert eine automatisierte Strukturierung der Anzeigen, in der die Bedeutung der einzelnen mittels Zeichensequenzen beschriebenen Informationen zu automatisiert zu erkennen ist.

4.2.2 *Impressumsseiten*

Nach deutscher Gesetzgebung [24] ist vorgesehen, dass Diensteanbieter einer Webpräsenz „den Namen und die Anschrift, unter der sie niedergelassen sind [...] unmittelbar erreichbar und ständig verfügbar zu halten [haben]“. Diese Informationen sind in Form des Impressums zu finden. Bei kommerziell betriebenen Webseiten lassen sich hier auch noch weitere verpflichtende Angaben finden. Eine Übersicht über regelmäßig auftretende Angaben ist in Tabelle 6 dargestellt. Die Präsentation der Angaben unterliegt keiner festen Formatvorgabe, häufig tauchen die Angaben jedoch in vorstrukturierter Form auf.

Die verfügbaren Informationen ließen sich bei einheitlicher Strukturierung zur automatisierten Erstellung und Aktualisierung von Unternehmensverzeichnissen nutzen. Weiterhin ist ein automatisiertes Einpflegen der identifizierten Unternehmensadressen in geographische Informationssysteme wie OpenStreetMap[11] oder Google Maps[12] vorstellbar.

11 http://www.openstreetmap.org, letzter Zugriff am 18.09.2015
12 http://maps.google.com, letzter Zugriff am 18.09.2015

Tabelle 6: Relevante Attribute in der Domäne *Impressumsseiten*

Attribut	Beschreibung
Name	Name des Webpräsenzbesitzers (natürliche Person, Unternehmen oder Institution)
Adresse	postalische Anschrift des Webpräsenzbesitzers
E-Mail-Adresse	E-Mail-Adresse zur Kontaktaufnahme zum Webpräsenzbesitzer
Telefonnummer	Telefonnnummer zur Kontaktaufnahme zum Webpräsenzbesitzer
Umsatzsteuer-identifikationsnummer	eindeutige Kennzeichnung des Unternehmens zu steuerlichen Zwecken

4.2.3 *Ausschreibungen studentischer Abschlussarbeiten*

Um Studierende auf mögliche Themen für Abschlussarbeiten hinzuweisen, hat es sich etabliert, dass konkrete Themenvorschläge vom wissenschaftlichen Personal der Hochschulen ausgeschrieben werden. Solche Ausschreibungen sind häufig in Papierform an den Pinnwänden der ausschreibenden Fachgebiete oder an zentralen Pinnwänden des Fachbereichs zu finden. Insbesondere werden die Ausschreibungen aber auch auf den Webseiten der einzelnen Fachgebiete veröffentlicht. Zentrale, hochschulweite Portale gibt es nur selten. Insbesondere diese Verteilung der Informationen erschwert die Suche für den Studierenden. So muss mangels zentraler Suche an den jeweiligen Stellen einzeln recherchiert werden. Ähnlich zur Domäne der Stellenanzeigen stellt sich auch hier die Herausforderung, dass für Studierende nicht immer offensichtlich ist, welcher Lehrstuhl Arbeiten ausschreibt, die zu ihren Interessen und ihren Voraussetzungen passen. Daher werden mitunter passende Ausschreibungen nicht aufgefunden. Teilweise haben Fachgebiete eigene Formatvorgaben für die Ausschreibungen, aber in der Regel gibt es kein fachgebietsübergreifendes Layout. Die in den Ausschreibungen aufzufindenden Informationen wiederholen sich jedoch häufig (siehe Tabelle 7).

Eine einheitlich strukturierte Darstellung der Ausschreibungen würde die Suche für Studenten vereinfachen. Insbesondere könnten beispielsweise universitätsweite Informationsportale für die Suche nach Abschlussarbeiten umgesetzt werden. Im Projekt *Die Masterarbeit*[13] wurde der Bedarf einer Suchmaschine für offene Abschlussarbeiten identifiziert, allerdings ist zum aktuellen Zeitpunkt nur eine Volltextsuche verfügbar.

4.2.4 *Kurzfassungen wissenschaftlicher Publikationen*

Wissenschaftliche Publikationen sind eine reiche Quelle an Informationen. In besonders komprimierter Form lassen sich Informationen in den Kurzfassungen der wissenschaftlichen Publikationen finden. Insbesondere lässt sich auf Basis dieser Kurzfassungen die Relevanz der Publikation für das aktuelle Informationsbedürfnis eines

13 http://www.die-masterarbeit.de, letzter Zugriff am 18.09.2015

Tabelle 7: Relevante Attribute in der Domäne *Ausschreibungen studentischer Abschlussarbeiten*

Attribut	Beschreibung
Titel	Titel der Ausschreibung
Beschreibung	ausführliche Beschreibung der Aufgabenstellung
Ziele	Ziele der Arbeit
Betreuer	Name des Betreuers der Arbeit
E-Mail	E-Mail-Adresse des Betreuers der Arbeit
Typ der Arbeit	Masterarbeit/Bachelorarbeit/Diplomarbeit/Studienarbeit
Vorkenntnisse/Voraussetzungen	Voraussetzungen, die der Interessent mitbringen sollte
Veröffentlichungsdatum	Datum der Veröffentlichung der Ausschreibung
Beginn	gewünschtes Startdatum für die Bearbeitung der Arbeit
Sprache	Sprache, in der die Ausarbeitung verfasst werden soll
Institutsbeschreibung	Beschreibung des ausschreibenden Fachgebiets

Suchenden identifizieren. Die Auswahl zu lesender Publikationen durch den Suchenden erfolgt aufgrund unterschiedlicher Kriterien, so können beispielsweise Publikationen gesucht werden, die das gleiche Ziel wie die eigene Forschung haben, oder Publikationen, die gleiche Techniken verwenden. Auch wenn die Kurzfassungen völlig unstrukturiert sind und aus reinem Fließtext bestehen, lassen sich gewisse Informationen mit einer hohen Regelmäßigkeit erkennen. Welche Attribute eine solche Kurzfassung repräsentieren können, lässt sich Tabelle 8 entnehmen.

Eine strukturierte Darstellung würde es ermöglichen, gezielt innnerhalb einzelner Attribute zu suchen, um Fragestellungen in der Forschung wie „Welche Veröffentlichungen haben eine ähnliche Motivation wie meine Arbeit?“ oder „Welche Arbeiten sind verwandt zu Methode xy?“ zu beantworten.

4.2.5 *Scrum-Protokolle*

Im Zuge der Durchsetzung agiler Methoden in der Softwareentwicklung hat sich Scrum als De-facto-Standard etabliert [169]. Ein wesentlicher Bestandteil von Scrum ist das regelmäßig stattfindende Retrospektivmeeting, an welchem das Scrum-Team und der Scrum-Master beteiligt sind. Hierbei berichten die Teammitglieder über die Erfahrungen aus dem vorigen Sprint[14]. Insbesondere wird während des Retrospektivmeetings darauf eingegangen, was zum Prozess positiv zu erwähnen ist, was negativ zu erwähnen ist und was Verbesserungspotenzial hat. Dies wird häufig in einem textuellen Protokoll festgehalten. Somit ergeben sich die Informationen, die in einem

14 Als Sprint wird eine definierte Zeitspanne, in der die Weiterentwicklung des Projektes vorangetrieben wird, bezeichnet.

solchen Protokoll zu finden sind (siehe Tabelle 9). Häufig sind in den Protokollen nur einzelne Stichpunkte enthalten.

Insbesondere zur Geschäftsprozessanalyse könnte die langfristige Auswertung dieser Protokolle einen Mehrwert liefern, da sich anhand derer elementare Stärken und Schwächen der Prozesse, eines Scrum Teams oder eines ganzen Unternehmens identifizieren ließen.

Tabelle 8: Relevante Attribute in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* [148]

Attribut	Beschreibung
Thema	Kategorisierung der Publikation
Zusammenfassung	sehr kurze Zusammenfassung der kompletten Publikation
Motivation	Motivation für den Inhalt der Publikation
Ziele	Ziele der Publikation
Methoden	technische Ansätze, die zur Umsetzung genutzt wurden
Verwandte Arbeiten	verwandte Arbeiten zur vorgestellten Arbeit
Lösungsansatz	Umsetzung zur Problemlösung
Ergebnisse	Ergebnisse, die erzielt wurden
Fazit	Fazit der Ergebnisse der Arbeit

Tabelle 9: Relevante Attribute in der Domäne *Scrum-Protokolle*

Attribut	Beschreibung
Anwesende	Namen der beteiligten Personen
Gut	positive Aspekte aus dem vergangenen Sprint
Schlecht	negative Aspekte aus dem vergangenen Sprint
Verbesserungswürdig	verbesserungswürdige Aspekte aus dem vergangenen Sprint

4.3 RELEVANTE ATTRIBUTTYPEN

Bei der Analyse der zuvor beschriebenen Domänen zeigt es sich, dass gewisse Typen von Attributen mehrfach wiederkehrend auftauchen. Die unterschiedlichen Typen unterscheiden sich in ihren Charakteristiken wesentlich und werden im Folgenden beschrieben. Eine Übersicht über das Auftauchen der Attributtypen in den unterschiedlichen, zuvor vorgestellten Domänen ist in Tabelle 10 zu finden.

Tabelle 10: Vorkommen der Attributtypen in den betrachteten Domänen, die Fußnoten annotieren jeweils die konkreten Attribute

Domäne	Eigennamen	Numerische Attribute	Freitext-attribute	Aggregierte Attribute	Meta-Attribute
Stellenanzeigen	X[a]	X[b]	X[c]	X[d]	X[e]
Impressumsseiten	X[f]	X[g]		X[h]	
Ausschreibungen studentischer Abschlussarbeiten	X[i]	X[j]	X[k]		X[l]
Kurzfassungen wissenschaftlicher Publikationen			X[m]		X[n]
Scrum-Protokolle			X[o]	X[p]	

a *Arbeitgeber, Ansprechpartner, Einsatzort*
b *Vergütung, Startdatum, Telefonnummer*
c *Titel, Anforderungen, Aufgaben, Arbeitgeberbeschreibung*
d *Anschrift*
e *Berufsfeld*
f *Name*
g *Umsatzsteuer-Identfikationsnummer, Telefonnummer*
h *Adresse*
i *Name des Betreuers*
j *Veröffentlichungsdatum*
k *Titel, Ziele, Beschreibung, Vorkenntnisse/Voraussetzungen, Institutsbeschreibung*
l *Themenbereich*
m *Zusammenfassung, Motivation, Ziele, Methoden, verwandte Arbeiten, Lösungsansatz, Ergebnisse, Fazit*
n *Thema*
o *Gut, Schlecht, Verbesserungswürdig*
p *Anwesende*

4.3.1 *Eigennamen*

Durch *Eigennamen* werden Namen beispielsweise von Firmen, Produkten, Orten oder Personen bezeichnet. In der Regel sind dieses einzelne Wörter oder kurze Sequenzen mehrerer Wörter, sogenannten Phrasen. So lassen sich beispielsweise der Einsatzort für eine ausgeschriebene Stelle oder der Name eines Webpräsenzinhabers durch ein Attribut des Typs Eigenname beschreiben.

4.3.2 *Numerische Attribute*

Numerische Attribute bestehen entweder aus rein numerischen Zeichenketten, wie zum Beispiel Postleitzahlen oder Jahresangaben, oder aus numerischen Zeichenketten in Kombination mit Sonderzeichen, wie beispielsweise Datumsangaben oder Telefonnummern.

4.3.3 *Freitextattribute*

Freitextattribute unterliegen keinen Formatbeschränkungen und zeichnen sich durch ihre große Heterogenität der internen Struktur aus. So können dies Stichpunkte, Halbsätze, ganze Sätze, Aufzählungen oder auch Absätze sein. So lassen sich beispielsweise Anforderungen für den Bewerber für eine Stelle in keinem standardisierten Format ausdrücken, weshalb dies ein Freitextattribut ist.

4.3.4 *Aggregierte Attribute*

Aggregierte Attribute bestehen aus Kaskadierungen mehrerer Attribute mitunter unterschiedlicher Attributtypen, wie zum Beispiel der zuvor beschriebenen atomaren Attributtypen. Sie zeichnen sich durch ein festes Muster aus, in welchem die atomaren Attribute, wie beispielsweise die in Abschnitt 4.3.1 bis 4.3.3 beschriebenen Attribute, angeordnet sind. Beispiele für aggregierte Attribute sind Adressen, bestehend aus den atomaren Attributen Straßenname (Eigenname), Hausnummer (numerisches Attribut), Postleitzahl (numerisches Attribut), Ort (Eigenname) und Land (Eigenname), die Liste der Namen der Teilnehmer eines Scrum-Retrospektivmeetings oder Produktbezeichnungen, bestehend aus dem Herstellernamen, dem Produktnamen und beispielsweise einer Modellnummer.

4.3.5 *Meta-Attribute*

Im Gegensatz zu den anderen zuvor genannten Attributen sind *Meta-Attribute* Zeichensequenzen, die nicht zwangsläufig in dieser Form im Dokument auftauchen müssen. Sie können dagegen aus dem textuellen Inhalt durch Hinzunahme von externem Wissen abgeleitet werden. Beispiele hierfür sind das Thema der Kurzfassung einer wissenschaftlichen Publikation oder das Berufsfeld für eine Stellenausschreibung.

4.4 DISKUSSION UND EIGENE BEITRÄGE

In diesem Kapitel wurde die Grundlage für die im Rahmen dieser Arbeit vorgestellten Verfahren zur Strukturierung von textuellen Dokumenten gelegt. Dazu wurde zunächst in die im Folgenden verwendete Terminologie eingeführt und die notwendigen Konzepte mittels eines Modells zueinander in Relation gesetzt. Weiterhin wurden fünf Anwendungsdomänen vorgestellt und betrachtet, welche Attributtypen in diesen Domänen auftauchen und wie sich diese charakterisieren lassen.

Auf Basis der als relevant identifizierten Attributtypen werden im Folgenden Verfahren zur Identifikation von Attributen dieser Typen vorgestellt. Der Bedarf an Forschung zur Identifikation der einzelnen Attribute hängt stark vom Attribut ab. Zur Erkennung von Eigennamen existieren bereits zahlreiche Ansätze (siehe Abschnitt 3.3.1). Insbesondere Verfahren, wie das von Notham et al. [119] vorgestellte, welche enzyklopädisches Wissen nutzen, lassen sich mit geringem Aufwand in neuen Domänen einsetzen. Numerische Attribute zeichnen sich durch ihre regelmäßige interne Struktur aus. So lassen sich numerische Attribute wie Telefonnummern oder Preise gut durch ihre Struktur charakterisieren und mittels dieser identifizieren. Freitextattribute zeichnen sich hingegen durch ihre sehr unregelmäßige interne und externe Struktur aus. Diese kann daher kaum zur Identifikation herangezogen werden. Aggregierte Attribute zeichnen sich durch ihre häufig regelmäßige interne Struktur aus, aber ihre Identifikation kann leicht fehlerbehaftet sein, da zur korrekten Identifikation alle atomaren Attribute korrekt identifiziert werden müssen. Weiterhin sind existierende Ansätze zur Identifikation spezifischer aggregierter Attribute wie Adressen nicht auf andere Domänen übertragbar. Zur Identifikation von Meta-Attributen bieten sich Textklassifikationsverfahren (siehe Abschnitt 2.2.2) an. Das Ziel der Domänenadaptivität von Textklassifikationsverfahren wurde in der Vergangenheit adressiert, allerdings haben die Verfahren spezifische Anforderungen bezüglich der Ähnlichkeit der zu nutzenden Klassen (siehe Abschnitt 3.4). Da solche Ähnlichkeiten zwischen den Werten der Meta-Attribute in unterschiedlichen Domänen nicht angenommen werden können, eignen sich diese Verfahren nicht zur domänenadaptiven Identifikation.

Auf Basis dieser Beobachtungen wird in den folgenden Kapiteln die Identifikation der Freitextattribute, der aggregierten Attribute und der Meta-Attribute im Detail betrachtet. Es werden Konzepte zu deren Identifikation vorgestellt, umgesetzt und evaluiert.

5

IDENTIFIKATION VON FREITEXTATTRIBUTEN

Freitextattribute unterliegen keinen Beschränkungen der Struktur (vergleiche Abschnitt 4.3.3). Aufgrund der heterogenen internen Struktur lassen sich zur Identifikation der Freitextattribute keine regelbasierten Ansätze nutzen. Auch die externe Struktur der Attribute lässt kein einheitliches Vorgehen zur Identifikation für Attribute vom Typ Freitextattribut zu. Daher wird in diesem Kapitel ein zweistufiges Verfahren zur Identifikation von Freitextattributen vorgestellt, welches individuell an eine Domäne anzupassen ist. Hierbei wird zunächst das Dokument in kleinere Segmente unterteilt und anschließend werden die einzelnen Segmente klassifiziert [148]. Die einzelnen Attribute stellen dabei die Zielklassen für die Klassifikation dar. Die Attributwerte sind die Textsegmente. Die Segmentierung erfolgt in Abhängigkeit der Domäne, der Entitätsausprägung und der damit verbundenen Attribute. Die Klassifikation erfolgt domänenunabhängig, es müssen nur domänenspezifische Trainingsdaten zur Verfügung stehen.

5.1 BESCHREIBUNG DES VERFAHRENS

Zunächst ist eine Segmentierung des zu strukturierenden Dokumentes vorzunehmen. Dabei sind die jeweiligen Eigenschaften eines konkreten Freitextattributes zu beachten. Während beispielsweise in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* (Abschnitt 4.2.4) jeder Satz einem anderen Attribut zugeordnet werden kann und hier eine Segmentierung auf Basis der Satzgrenzen notwendig ist, ist in der Domäne *Stellenanzeigen* (Abschnitt 4.2.1) davon auszugehen, dass beispielsweise die Arbeitgeberbeschreibung aus einem kompletten Absatz besteht, so dass eine Segmentierung auf Basis der Absatzgrenzen durchgeführt werden sollte.

Bei der Anwendung von Klassifikationsverfahren sind allgemein zwei grundsätzliche Designentscheidungen zu treffen (siehe Abschnitt 2.2.1). Dies ist zum einen die Wahl der Merkmale, die die zu klassifizierenden Instanzen beschreiben, und zum anderen die konkrete Wahl der verwendeten Klassifikationsverfahren. Im Folgenden werden diese beiden Aspekte unter dem Gesichtspunkt, Klassifikationsverfahren zur Identifikation von Freitextattributen zu verwenden, diskutiert.

5.1.1 *Merkmalsgruppen*

Zunächst werden die Merkmale beschrieben, welche genutzt werden, um die einzelnen Instanzen zu repräsentieren. Diese Repräsentation wird zur Klassifikation verwendet, die das Ziel hat den Textsegmenten ein Attribut zuzuordnen, so dass die Textsegmente als Attributwerte identifiziert werden. Für jede Instanz wird unter Nutzung dieser Merkmale ein Merkmalsvektor gebildet, welcher vom Klassifikator genutzt wird. Die Merkmale sind in diesem Kapitel als Merkmalsgruppen zusammengefasst. So umfasst beispielsweise die Merkmalsgruppe *Unigramme* (Abschnitt

5.1.1.1) für jedes Unigramm ein einzelnes Merkmal. Die Auswahl der Merkmale wurde unter Berücksichtigung des Standes der Forschung im Bereich der Textklassifikation durchgeführt (siehe Abschnitt 2.2.2).

5.1.1.1 *Unigramme*

Als grundlegende Merkmale zur Klassifikation von Texten haben sich Unigramme als geeignet erwiesen (vergleiche Abschnitt 2.2.2). Diese werden im hier vorgestellten Verfahren verwendet. Es werden die 1.000 im Trainingskorpus am häufigsten auftauchenden Unigramme als Merkmale genutzt. Um der unterschiedlichen Häufigkeit und Relevanz der Unigramme gerecht zu werden, wird eine Gewichtung anhand des TF-IDF-Schemas vorgenommen.

5.1.1.2 *Sentiment*

Durch das Sentiment eines Satzes wird die emotionale Haltung des Verfassers zum Geschriebenen ausgedrückt. Da diese Haltung häufig mit dem Inhalt des Textes korreliert, wird das Sentiment als Merkmal verwendet. Sentimenterkennung ist ein eigenständiger Forschungsbereich im Bereich des Natural Language Processings [91]. In dem vorgestellten Verfahren wird ein listenbasierter Ansatz zur Sentimenterkennung genutzt. Konkret wird hierzu das SentiWordNet [7] verwendet, welches aus einer Abbildung von Wörtern zu Sentimentwerten besteht. Die Werte wurden per semi-überwachtem Lernverfahren bestimmt und liegen im Intervall $[-1,0;1,0]$, wobei ein Wert > 0 für ein positives Sentiment steht und ein Wert < 0 ein negatives Sentiment repräsentiert. Basierend auf dieser Liste werden für jede Instanz drei konkrete Merkmale abgeleitet: die Summe aller Sentimentwerte für Wörter aus der entsprechenden Instanz, die Summe aller positiven Sentimentwerte für Wörter der Instanz und die Summe aller negativen Sentiment-Werte der Instanz.

5.1.1.3 *Negation*

In Ergänzung zum Sentiment werden Negationen in der zu klassifizierenden Instanz betrachtet. Dies wird insbesondere deswegen gemacht, um Negationen der Sentimente identifizieren zu können. Zur Erkennung der Negation wird ein Abgleich mit einer Wortliste[1] vorgenommen. Der Wert des Merkmals ergibt sich durch die Häufigkeit des Auftauchens von Token aus der Wortliste in der jeweiligen Instanz.

5.1.1.4 *Zeit*

Die verwendete Zeitform eines Textabschnitts kann Details über die enthaltene Information vermitteln, da beispielsweise die Werte gewisser Attribute eher in der Vergangenheitsform verfasst sind. Durch Nutzung eines *lexikalischen Parsers* kann die grammatikalische Struktur eines Satzes und somit auch die Zeitform identifiziert werden. Die Anzahl der Verben in Gegenwartsform und Vergangenheitsheitsform werden in zwei unabhängigen Merkmalen repräsentiert. Für die Anzahl der Verben in Gegenwartsform wird die Häufigkeit der POS-Tags *VBP*, *VBZ* und *VBG* gezählt, für die Anzahl der Verben in Vergangenheitsform dagegen die Häufigkeit der POS-Tags *VBD* und *VBN* (siehe Tabelle 31 im Anhang A.1 für die Erklärung der Tags).

1 Die Wortliste für die englische Sprache enthält die Token „not“, „never“ und „no“.

5.1.1.5 *Zeitindikatoren*

Insbesondere bei unvollständigen Sätzen liefern lexikalische Parser häufig inkorrekte Ergebnisse. Weiterhin liegen Sprachmodelle häufig nur für eine begrenzte Zahl an Sprachen vor[2]. Aus diesem Grund werden zwei heuristische Merkmale zur Identifikation der Zeitform hinzugefügt. Eines der Merkmale repräsentiert die Häufigkeit von Suffixen, die auf eine Vergangenheitsform hindeuten[3], das andere Merkmal repräsentiert die Häufigkeit von Hilfsverben, die auf eine Vergangenheitsform hindeuten[4].

5.1.1.6 *Adjektive*

Adjektive stellen eine zentrale Rolle bei der Interpretation des Inhaltes eines Textes dar. Die Anzahl der Adjektive kann durch die Nutzung des lexikalischen Parsers ermittelt werden, indem die Häufigkeit der POS-Tags *JJ*, *JJR* und *JJS* gezählt wird (siehe Tabelle 31 für die Erklärung der Tags).

5.1.1.7 *Imperativindikatoren*

Gewisse Formen von Informationen sind charakterisiert durch die Verwendung des Imperativs. Auf Basis einer Liste von Token, die Indikatoren für den Imperativ sind[5], enthält der Merkmalsvektor einen Eintrag für die summierte Häufigkeit dieser Token in der Instanz.

5.1.1.8 *Personalpronomen*

Anhand der Personalpronomen lassen sich Informationen über Verfasser und Subjekt eines Textes bestimmen. Die Häufigkeit der Personalpronomen, bestimmt durch den lexikalischen Parser, wird als weiteres Merkmal genutzt. Dazu wird die Häufigkeit des Auftauchens des POS-Tags *PRP* gezählt (siehe Tabelle 31 für die Erklärung der Tags).

5.1.1.9 *Position*

Die Position eines zu klassifizierenden Textsegments innerhalb eines gesamten Dokumentes spiegelt gewisse Hinweise über den Typ der Information wider. Gewisse Attribute tauchen eher am Anfang eines Dokumentes auf, andere Informationen eher im Mittelteil oder am Ende. Um den Einfluss der unterschiedlichen Dokumentenlängen gering zu halten, wird die relative Position als Merkmal genutzt. Hierzu wird die absolute Position durch die Gesamtzahl der zu klassifizierenden Textsegmente dividiert.

---

2 Eine Übersicht über verfügbare Sprachmodelle für den im Rahmen dieser Arbeit verwendeten Stanford Lexicalized Parser lässt sich unter `http://nlp.stanford.edu/software/lex-parser.shtml` finden.

3 Im Englischen wird auf den Wort-Suffix „ed“ geprüft.

4 Im Englischen wird die Liste „had“, „was“, „were“, „been“ und „got“ verwendet.

5 Die Liste mit englischen Indikatoren für den Imperativ enthält die Terme „need“, „should“ und „must“.

5.1.1.10 *Tokenanzahl*

Aus der Länge eines zu klassifizierenden Textsegments können Rückschlüsse über den Typ der Information geschlossen werden, da die Werte gewisser Attribute eher länger als anderer Attribute sind. Neben der Gesamtzahl an Token in einer zu klassifizierenden Tokensequenz wird die Anzahl numerischer Token als Merkmal genutzt.

5.1.2 *Klassifikationsverfahren*

Wie zuvor beschrieben, haben unterschiedliche Lernverfahren ihre jeweiligen Stärken und Schwächen (siehe Abschnitt 2.2.1). Um einen eventuellen Einfluss der Wahl des konkreten überwachten Lernverfahrens erkennen zu können, werden drei verschiedenen Klassen überwachter Lernverfahren aufgrund ihrer Vorteile eingesetzt:

- Support Vector Machines (SVMs) haben in der Vergangenheit durch sehr hohe Güte bei der Textklassifikation überzeugt.
- Bayes-Klassifikatoren zeichnen sich durch eine gute Performanz bei sehr hoher Güte aus.
- Entscheidungsbäume stechen durch ihre intuitive Verständlichkeit bei hoher Güte hervor.

5.2 EVALUATION DES VERFAHRENS

Um Aussagen über die Güte des vorgestellten Ansatzes treffen zu können, erfolgt eine Evaluation. Nachfolgend wird zunächst auf die verwendeten Daten für die Evaluation eingegangen, gefolgt von einer Erläuterung des Evaluationsaufbaus. Anschließend werden die Ergebnisse der Evaluation vorgestellt und diskutiert. Die Klassenlabel der einzelnen Instanzen repräsentieren die Attribute, wohingegen die Instanzen selbst die Attributwerte darstellen.

5.2.1 *Evaluationsdaten*

Das beschriebene Verfahren wird in zwei der vorgestellten Domänen (siehe Abschnitt 4.2) evaluiert. Als Goldstandard werden jeweils Evaluationskorpora aus diesen Domänen verwendet.

5.2.1.1 *Kurzfassungen wissenschaftlicher Publikationen*

In der Domäne *Kurzfassungen wissenschaftlicher Publikationen* wird eine Segmentierung auf Satzebene vorgenommen. Es wird davon ausgegangen, dass jeder Satz ein neues Segment repräsentiert. Zur Evaluation des Verfahrens in dieser Domäne werden zwei verschiedene englischsprachige Korpora genutzt. Die Korpora repräsentieren unterschiedliche Entitätsausprägungen der Domäne, da sie jeweils unterschiedliche Freitextattribute enthalten. Für den Korpus *MM (Multimedia)* wurden 81 Kurzfassungen wissenschaftlicher Publikationen zusammengetragen, welche in englischer Sprache verfasst sind. Die Kurzfassungen wurden aus der ACM Digital

Library der *ACM Transactions on Multimedia Computing, Communications and Applications*[6] Bände 7 bis 9 entnommen. Insgesamt bestehen diese 81 Kurzfassungen aus 628 Sätzen. Diese Sätze wurden manuell von drei Annotatoren annotiert. Hierzu wurden die in Tabelle 8 in Kapitel 4 dargestellten Label, die den durch das Verfahren zu bestimmenden Attributen entsprechen, genutzt. Das Attribute *Thema* wurde nicht zur Evaluation verwendet, da dies ein Meta-Attribut ist, welches für die komplette Kurzfassung gültig ist und nicht nur für einen einzelnen Satz. Tabelle 11 zeigt die Beschreibungen, die die einzelnen Annotatoren für die Label erhalten haben. Die Annotatoren erhielten die Instruktion, wenn möglich genau ein Klassenlabel pro Satz zu vergeben; falls ein Satz jedoch eindeutig zwei oder mehr Klassen zuzuordnen ist, konnten auch mehrere Klassenlabel für einen Satz vergeben werden.

Tabelle 11: Verwendete Label für den Korpus *MM* und deren Beschreibung

Label	Beschreibung
Zusammenfassung	eine komplette Zusammenfassung der Arbeit, meist in einem einzelnen Satz
Motivation	Was ist die Motivation für die vorliegende Arbeit? Wieso ist die Arbeit relevant? Was ist die Herausforderung?
Ziele	Welches sind die Ziele der Arbeit?
Methoden	Welche Ansätze wurden zur Problemlösung genutzt? (Techniken, Schritte im Designprozess et cetera)
Verwandte Arbeiten	Was wurde im Forschungsgebiet bisher gemacht? Wo gab es Erfolge? Wo gab es Misserfolge?
Lösungsansatz	Wie funktioniert der vorgestellte Ansatz? Was ist die zugrunde liegende Idee?
Ergebnisse	Wie fallen die Evaluationsergebnisse aus?
Fazit	Ist das Ergebnis zufriedenstellend? Wie kann es genutzt werden?

Tabelle 12 zeigt die Häufigkeiten der einzelnen Label mit denen die Label durch die einzelnen Annotatoren gewählt wurden sowie die Häufigkeit der vollständigen Übereinstimmungen und der mehrheitlichen Übereinstimmungen, bei denen zwei der drei Annotatoren das gleiche Label wählten. Eine vollständige Übereinstimmung wurde für 40,8% der Satz-Label-Paarungen erzielt und eine mehrheitliche Übereinstimmung für 86,7% der Paarungen. Auf Basis der hohen mehrheitlichen Übereinstimmung werden zur Evaluation die Sätze verwendet, bei denen mit mehrheitlicher Übereinstimmung ein Klassenlabel gewählt wurde. Nach Entfernen der Sätze, denen mehrheitlich mehrere Label zugeordnet wurden, besteht der Evaluationskorpus *MM* aus 546 Sätzen.

Der im Folgenden als *BioMed* bezeichnete Korpus [60] besteht aus 1.000 englischsprachigen Kurzfassungen wissenschaftlicher Publikationen mit insgesamt 8.633 Sätzen. Die Publikationen stammen alle aus dem biomedizinischen Forschungsbereich und sind satzweise mit einem der folgenden sieben Klassenlabel versehen: *Hinter-*

6 http://dl.acm.org/citation.cfm?id=J961, letzter Zugriff am 17.10.2015

Tabelle 12: Häufigkeit der einzelnen Klassenlabel bei der Annotation des Korpus *MM*

Label	Annotator			Übereinstimmung	
	1	2	3	Vollständig	Mehrheitlich
Zusammenfassung	74	69	141	51	73
Motivation	163	191	140	100	165
Ziele	22	11	22	2	11
Methoden	95	34	125	0	25
Verwandte Arbeiten	82	41	82	20	64
Lösungsansatz	203	239	48	27	162
Ergebnisse	99	96	76	56	87
Fazit	25	9	17	0	8
Insgesamt (628)	763	690	651	256	595

grund, Ergebnisse, Methode, Fazit, Ziel, Verwandte Arbeiten und *Zukünftige Arbeiten*. Zur Erstellung des Datensatzes durch Guo et al. [60] wurde zunächst ein kleiner Teildatensatz durch drei Annotatoren annotiert. Wegen der hohen Interrater-Reliabilität ($\kappa = 0,85$) wurde zur Minimierung des Annotationsaufwands für den kompletten Datensatz auf die Annotationen eines einzelnen Annotators zurückgegriffen.

5.2.1.2 *Scrum-Protokolle*

Zur Evaluation in der zweiten Anwendungsdomäne wurde ein Korpus bestehend aus 139 englischsprachigen Protokollen von Scrum-Retrospektivmeetings verwendet. Die Protokolle bestehen zu einem großen Teil aus mehreren Absätzen mit stichpunktartigen Aufzählungen, welche gute, schlechte und verbesserungswürdige Stichpunkte des vergangenen Sprints beschreiben. Jeder der Stichpunkte wird als ein eigenständiges Segment angenommen, welches zu klassifizieren ist. Zur Erstellung des Evaluationskorpus wurden die Stichpunkte in den einzelnen Absätzen nach diesen Punkten gruppiert, diese Gruppierung wurde auf Basis der jeweiligen Absatzüberschrift vorgenommen. Dabei wurden Absätze mit semantisch ähnlicher Überschrift zusammengefasst[7]. Dieses Verfahren ergibt insgesamt 653 Stichpunkte, die den drei Klassenlabeln *gut, schlecht* und *verbesserungswürdig* zugeordnet werden können. Die Gesamtzahl der annotierten Stichpunkte ist im Korpus *Scrum* zusammengefasst. Weiterhin wurde eine Teilmenge des Korpus ausgewählt, welche nur aus Stichpunkten besteht, deren eindeutige Zuordnung zu einem der Label möglich ist[8]. Diese Teilmenge, welche im Folgenden als *ScrumSubset* benannt wird, umfasst 442 annotierte Stichpunkte.

7 Beispielsweise werden Absätze mit den Überschriften „What should we change?“, „What should we do differently?“ und „What can we improve?“ zusammengefasst.

8 Stichpunkte wie „Slow start“ oder „Better collaboration would increase efficiency“ gelten als eindeutig zuordnungsbar, wohingegen „Timing“ oder „Collaboration“ nicht eindeutig einer Klasse zuzuordnen sind.

5.2.2 *Evaluationsmethodik*

Das beschriebene Verfahren zur Identifikation von Freitextattributen wurde in Java implementiert. In der Implementierung wird das *Waikato Environment for Knowledge Analysis (Weka)* [63] als Machine Learning Framework genutzt. Insbesondere werden aus dem Weka Framework die Implementierungen der einzelnen Klassifikatoren genutzt: *Sequential Minimal Optimization (SMO)* [78, 127] als Vertreter der SVMs, *Naive Bayes Klassifikator (NB)* [74] als Vertreter der Bayes-Klassifikatoren und *J48* [129] als Vertreter der Entscheidungsbäume. Weiterhin werden zur Bestimmung der Werte der grammatikalischen Merkmale der *Stanford Lexicalized Parser* [35] genutzt.

Um Effekte einer unrepräsentativen Aufteilung in Trainings- und Testdaten zu verhindern, wird eine 10-fach stratifizierte Kreuzvalidierung durchgeführt. Für jeden der zehn Evaluationen werden die Ergebnisse für die einzelnen Klassen per micro-averaging arithmetisch gemittelt (siehe Abschnitt 2.2.3).

5.2.3 *Ergebnisse*

Im Folgenden werden die Evaluationsergebnisse für die beiden Domänen *Kurzfassungen wissenschaftlicher Publikationen* und *Scrum-Protokolle* vorgestellt.

5.2.3.1 *Kurzfassungen wissenschaftlicher Publikationen*

Tabelle 13 zeigt die Ergebnisse der Evaluation des Verfahrens unter Nutzung aller Merkmale sowie aller Merkmale unter Ausschluss einer einzelnen Merkmalsgruppe in der Domäne *Kurzfassungen wissenschaftlicher Publikationen*. Dargestellt sind die Ergebnisse für beide in dieser Domäne verwendeten Korpora und die drei verwendeten Klassifikatoren.

Bei Verwendung aller Merkmale zeigen sich die besten Ergebnisse unter Nutzung der SVM, wobei die Ergebnisse für den Korpus *BioMed* um 10,6 Prozentpunkte besser sind als für den Korpus *MM*. Bei Entfernen der *Unigram*-Merkmale ist über alle Kombinationen von Klassifizieren und Korpora der deutlichste Abfall des F1-Maßes zu erkennen. Unter Verwendung der SVM ist ebenso bei Entfernung des *Positions*-Merkmals eine deutliche Verschlechterung der Ergebnisse zu sehen. Das Entfernen der anderen Merkmale hat dagegen nur unwesentlichen Einfluss auf die Ergebnisse. Eine weitergehende Analyse der Merkmale hat die hohe Relevanz des Merkmals *Position* bestätigt. Es hat sich jedoch auch gezeigt, dass bei Betrachtung einzelner Merkmale, und nicht kompletter Merkmalsgruppen wie in diesem Kapitel, nicht nur *Unigramm*-Merkmale unter den relevantesten Merkmalen sind (siehe Anhang A.2). Interessanterweise verbessern sich die Ergebnisse durch das Entfernen einzelner Merkmale (*Adjektive* beziehungsweise *Sentiment*) jedoch leicht. Die Güte bei der Verwendung einer einzelnen Merkmalsgruppe zur Klassifikation ist Tabelle 14 zu entnehmen.

Bei Analyse des Korpus *MM* unter Verwendung einer einzelnen Merkmalsgruppe, schneidet bei den meisten Merkmalsgruppen der NB-Klassifikator am besten ab. Bei Verwendung des Korpus *BioMed* zeigen SVM beziehungsweise NB je nach Verwendung der Merkmale die besten Ergebnisse. Vergleichend lässt sich beobachten, dass sich die Ergebnisse bei Verwendung aller Merkmale ($F_1 = 69,2\%$ für *MM* beziehungsweise $F_1 = 79,8\%$ für *BioMed*) verglichen mit der reinen Verwendung

Tabelle 13: F1-Maß (in %) für die unterschiedlichen Klassifikatoren und Korpora in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* bei Verwendung aller Merkmale sowie aller Merkmale außer einer einzelnen Merkmalsgruppe

	MM			*BioMed*		
MERKMALE	SVM	NB	J48	SVM	NB	J48
ALLE	69,2	69,0	64,0	79,8	73,1	73,9
ALLE AUSSER						
Unigramme	55,5	49,2	51,0	66,6	60,5	64,8
Sentiment	69,4	71,0	64,4	80,0	73,2	73,9
Negation	68,8	69,3	64,1	79,9	73,4	74,1
Zeit	68,6	69,9	64,1	79,8	73,0	73,7
Zeitindikatoren	69,2	69,7	64,1	79,9	73,6	73,8
Adjektive	69,9	69,2	64,1	79,9	73,5	73,8
Imperativindikatoren	68,9	69,0	64,1	79,9	73,1	74,2
Personalpronomen	69,0	69,0	64,1	79,8	73,1	73,9
Position	63,4	65,6	57,6	75,0	67,0	67,5
Tokenanzahl	69,2	69,4	63,9	79,9	74,1	74,1

der *Unigramme* ($F_1 = 63,4\%$ für *MM* beziehungsweise $F_1 = 74,8\%$ für *BioMed*) um 5,8 Prozentpunkte (bei Verwendung von *MM*) beziehungsweise 5,0 Prozentpunkte (bei Verwendung von *BioMed*) verbessern, wenn SVMs genutzt werden. Auch bei Verwendung der anderen Klassifizierer lässt sich eine Verbesserung durch die Hinzunahme der weiteren Merkmale zusätzlich zu den sonst meist einzeln verwendeten *Unigrammen* erzielen. Abbildung 7 zeigt die Güte des Verfahrens bei Verwendung des Korpus *BioMed* in unterschiedlichem Umfang. Hierbei wurden unterschiedlich große, zufällig ausgewählte Teilmengen des Korpus zur 10-fach stratifizierten Kreuzvalidierung einer SVM verwendet. Es zeigt sich, dass sich ab der Verwendung von 2.000-3.000 Instanzen eine relativ konstante Güte herausbildet, es also zu einer Sättigung kommt, jedoch bei weniger Instanzen ein deutlicher Abfall der Güte zu erkennen ist. Weiterhin ist zu erkennen, dass die Klassfikationsgüte für den Korpus *BioMed* bei einer äquivalenten Größe zum Korpus *MM* sehr ähnlich zur Klassfikationsgüte bei diesem Korpus ($F_1 = 69,2\%$ für *MM* gegenüber $F_1 \approx 70\%$ für *BioMed* bei gleicher Größe) ausfällt. Dies lässt darauf schließen, dass die beiden verwendeten Korpora trotz unterschiedlicher Label-Mengen recht ähnliche Eigenschaften aufweisen und somit hier die jeweilige Entitätsausprägung keinen Einfluss auf die Klassifikationsgüte hat.

5.2.3.2 *Scrum-Protokolle*

Die Ergebnisse der Evaluation in der Domäne der *Scrum-Protokolle* unter Nutzung aller Merkmale und aller Merkmale unter Ausschluss einzelner Merkmalsgruppen ist Tabelle 15 zu entnehmen. Zu beachten ist, dass für die Evaluationen in dieser Anwendungsdomäne das Merkmal *Position* nicht verwendet wurde, da in den verwendeten

Tabelle 14: F1-Maß (in %) für die unterschiedlichen Klassifikatoren und Korpora in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* bei Verwendung einzelner Merkmalsgruppen

	MM			*BioMed*		
MERKMALE	SVM	NB	J48	SVM	NB	J48
NUR						
Unigramme	63,4	66,8	57,5	74,8	68,3	66,8
Sentiment	14,1	22,4	22,3	24,7	26,2	33,8
Negation	14,8	16,2	15,4	24,7	24,7	24,7
Zeit	24,8	24,3	24,2	32,3	32,6	35,8
Zeitindikatoren	27,8	27,9	26,5	25,4	31,9	31,9
Adjektive	16,6	16,8	19,3	24,7	25,0	24,7
Imperativindikatoren	15,5	14,7	14,3	25,4	25,0	25,4
Personalpronomen	27,4	26,9	26,8	27,9	28,0	28,0
Position	48,9	48,7	49,2	55,7	54,0	55,4
Tokenanzahl	14,3	24,1	25,7	24,8	27,6	30,0

Korpora die zu klassifizierenden Instanzen bereits nach Klasse sortiert waren und die Position innerhalb der Originaldokumente nicht verfügbar war. Somit hätte eine Nutzung dieses Merkmals unrepräsentativ gute Evaluationsergebnisse geliefert. Bei Einbindung aller Merkmale werden die besten Ergebnisse bei Verwendung der SVM beziehungsweise des NB-Klassifikators erzielt. Die Klassifikationsgüte ist für den Korpus *ScrumSubset* höher als für den Korpus *Scrum*. Bei Entfernung einzelner Merkmalsgruppen zeigt sich, dass die *Unigramm*-Merkmale den größten Einfluss auf die Klassifikationsgüte haben. Weiterhin lässt sich ein leichter Abfall der Klassifikationsgüte bei Entfernung der *Sentiment*-Merkmalsgruppe feststellen. Dieser Einfluss der Merkmalsgruppe *Sentiment* wird durch die Betrachtung des Informationsgewinns bestätigt (siehe Tabelle 35 in Anhang A.2). Auch bei Betrachtung der Ergebnisse für die Verwendung einer einzelnen Merkmalsgruppe (vergleiche Tabelle 16) lässt sich die beste Klassifikationsgüte für die Merkmalsgruppe *Unigramme* feststellen. In allen Evaluationen in dieser Domäne ist weiterhin der Einfluss der Merkmalsgruppe *Sentiment* bei Verwendung des J48-Klassifikators bemerkenswert. So sorgt sowohl das Entfernen dieser Merkmalsgruppe (vergleiche Tabelle 15) für eine deutliche Verschlechterung der Klassifikationsgüte als auch das alleinige Verwenden dieser Merkmalsgruppe (vergleiche Tabelle 16) für verhältnismäßig gute Ergebnisse.

Insgesamt ist festzustellen, dass die Ergebnisse in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* besser ausfallen als in der Domäne *Scrum-Protokolle*, obwohl in der zweiten Domäne eine kleinere Menge an Klassen vorliegt. Auch die unterschiedlichen Korpusgrößen haben hier keinen Einfluss, da selbst bei Verkleinerung des Korpus *BioMed* zur gleichen Größe wie der Korpus *Scrum*, die Ergebnisse für *BioMed* ($F_1 \approx 72\%$) besser ausfallen als für *Scrum* ($F_1 \approx 57\%$).

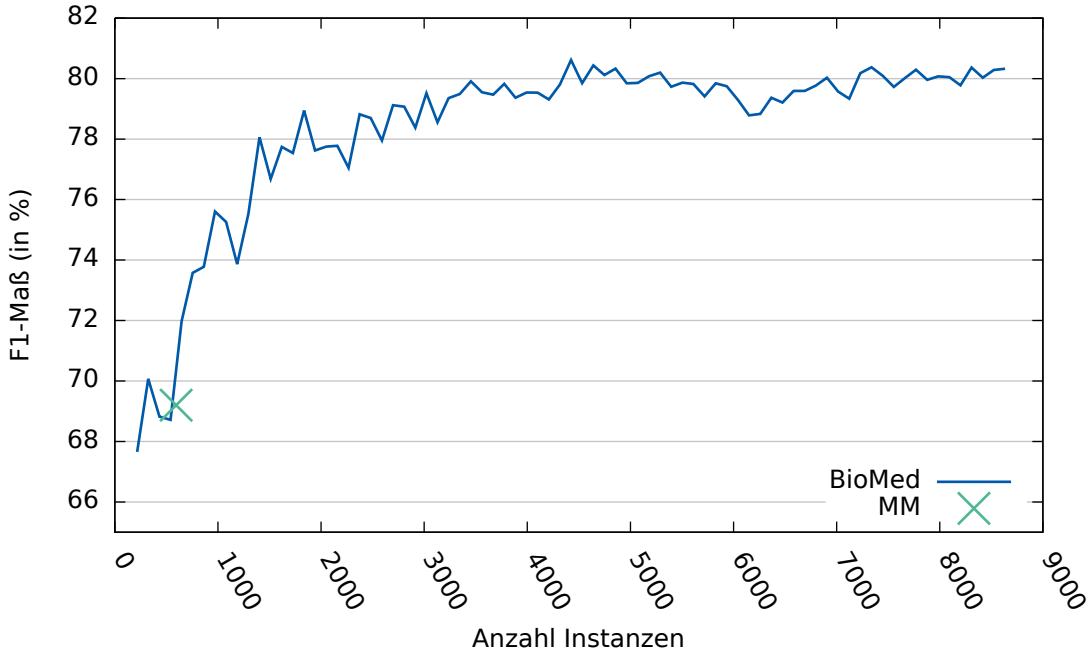


Abbildung 7: Lernkurve des F1-Maßes unter Nutzung des Korpus *BioMed* im Vergleich zum Korpus *MM*

5.3 FAZIT

In diesem Kapitel wurde ein Verfahren zur Identifikation von Freitextattributen vorgestellt. Das Verfahren basiert auf einer Segmentierung der Dokumente mit anschließender Klassifikation der einzelnen Segmente. Der vorgestellte Klassifikationsansatz nutzt keine domänenspezifischen Merkmale, so dass eine Übertragung in andere Anwendungsdomänen gut möglich ist. Als geeignetste Merkmale haben sich wie in verwandten Arbeiten Unigramme, also die im Text enthaltenen Wörter, herausgestellt. Dies erlaubt die Identifikation von Freitextattributen, deren Attributwert eine für das Attribut spezifische Wortverteilung enthält. Das vorgestellte Verfahren wurde in zwei Anwendungsdomänen evaluiert. Für die verwendeten Evaluationskorpora konnte gezeigt werden, dass innerhalb einer Anwendungsdomäne die Klassifikationsgüte stark von der Anzahl der verfügbaren Trainingsdaten abhängt, aber bei gleicher Anzahl unabhängig von der verwendeten Entitätsausprägungen eine sehr ähnliche Klassifikationsgüte erreicht wird. Weiterhin wurde gezeigt, dass die Klassifikationsgüte bei unterschiedlichen Domänen selbst bei gleicher Korpusgröße unterschiedlich ausfällt. Demnach ist die Identifikation von Freitextattributen für unterschiedliche Entitäten beziehungsweise Domänen unterschiedlich komplex.

Tabelle 15: F1-Maß (in %) für die unterschiedlichen Klassifikatoren und Korpora in der Domäne der *Scrum-Protokolle* bei Verwendung aller Merkmale sowie aller Merkmale außer einer einzelnen Merkmalsgruppe

	Scrum			*ScrumSubset*		
MERKMALE	SVM	NB	J48	SVM	NB	J48
ALLE	57,2	56,2	51,3	66,1	66,9	59,2
ALLE AUSSER						
Unigramme	46,7	48,4	46,6	55,0	57,0	54,8
Sentiment	55,8	55,0	49,5	65,6	65,0	56,5
Negation	57,4	56,0	50,6	66,1	68,8	60,0
Zeit	56,5	56,7	52,2	67,3	66,9	60,3
Zeitindikatoren	56,3	55,6	50,3	65,5	66,6	58,8
Adjektive	57,2	56,0	52,0	66,4	68,5	60,6
Imperativindikatoren	56,9	56,5	51,1	65,7	67,1	59,0
Personalpronomen	57,4	56,5	51,5	66,3	67,3	59,3
Tokenanzahl	56,7	56,0	51,0	65,3	68,0	60,5

Tabelle 16: F1-Maß (in %) für die unterschiedlichen Klassifikatoren und Korpora in der Domäne der *Scrum-Protokolle* bei Verwendung einzelner Merkmalsgruppen

	Scrum			*ScrumSubset*		
MERKMALE	SVM	NB	J48	SVM	NB	J48
NUR						
Unigramme	55,2	53,3	48,5	64,7	64,4	54,6
Sentiment	32,3	37,9	42,5	41,5	46,4	45,8
Negation	29,3	30,9	30,9	36,4	36,4	36,4
Zeit	38,2	39,2	39,9	26,0	34,6	46,2
Zeitindikatoren	35,7	33,9	41,0	36,6	36,6	31,5
Adjektive	31,0	33,6	34,0	34,6	36,2	34,1
Imperativindikatoren	34,1	34,1	34,1	39,1	39,2	39,2
Personalpronomen	33,2	32,2	32,5	28,5	34,9	33,2
Tokenanzahl	32,4	37,5	41,0	28,0	37,6	36,3

6

IDENTIFIKATION VON META-ATTRIBUTEN

Meta-Attribute sind Attribute, die aus dem Inhalt eines Dokuments abgeleitet werden. Die jeweiligen Attributwerte sind nicht notwendigerweise wörtlich im Dokument zu finden (vergleiche Abschnitt 4.3). Es bieten sich Klassifikationsverfahren zur Identifikation der Meta-Attribute an. Diese erlauben es, Informationen aus Texten auf Basis ihres Inhaltes abzuleiten. Da die Klassen, welche den Attributwerten der Meta-Attribute entsprechen, vorab definiert und damit bekannt sind, eignen sich überwachte Ansätze des maschinellen Lernens als Klassifikationsverfahren (vergleiche Abschnitt 2.2.1). Überwachte maschinelle Lernverfahren benötigen vorab annotierte Trainingsdaten zur Klassifikation. Da Attribute und Trainingsdaten domänenspezifisch sind, ist ein Domänenwechsel mit der Notwendigkeit der Annotation von Trainingsdaten für die Zieldomäne verbunden. Diese Annotation stellt eine der größten Hürden für die Nutzung von überwachten maschinellen Lernverfahren dar. Existierende domänenadaptive Klassifikationsverfahren haben hohe Anforderungen bezüglich der Ähnlichkeiten der Klassen in Ausgangs- und Zieldomäne (vergleiche Abschnitt 3.4). Aus diesen Gründen sind Verfahren mit geringen Anforderungen an die Anzahl an Trainingsdokumenten bei hoher Klassifikationsgüte anderen Verfahren gegenüber zu bevorzugen. Eine geringere Menge zu annotierender Trainingsdaten für eine neue Domäne vereinfacht die Adaption des Verfahrens an diese neue Domäne. In diesem Kapitel wird mit dem *Combined Ensemble and Fast Active Learner (CENFA)* ein solches Verfahren vorgestellt und die Evaluation in einer der Anwendungsdomänen präsentiert [149, 150].

6.1 GRUNDLAGEN DES VERFAHRENS

Das vorgestellte Verfahren CENFA basiert auf zwei Konzepten des maschinellen Lernens: *Ensemble Learning* und *Active Learning*. Während ersteres auf eine hohe Klassifikationsgüte und einen robusten Klassifikator abzielt, ist das Ziel des Active Learning mit weniger Trainingsdaten als beim klassischen überwachten Lernen eine mindestens gleich gute Klassifikationsgüte zu erreichen. Im Folgenden wird auf diese beiden Konzepte eingegangen.

6.1.1 *Ensemble Learning*

Ensemble Learning beschreibt eine Menge von Verfahren des maschinellen Lernens, welche mit konkreten Klassifikatoren (siehe Abschnitt 2.2.1) kombiniert werden können, um einen kombinierten Klassifikator zu konstruieren. Das Ziel der Verfahren des Ensemble Learning ist im Gegensatz zu den klassischen Klassifikatoren des überwachten Lernens nicht das Finden einer einzelnen Hypothese h, sondern einer Menge von k Hypothesen $\{h_1, \ldots, h_k\}$, welche durch Aggregation eine Klassifikationsentscheidung treffen. Die aggregierte Hypothese h^* wird gebildet durch Linear-

kombination der einzelnen Hypothesen mit Gewichtungsfaktoren $\{w_1, \ldots, w_k\}$ [38], also

$$h^*(x) = w_1 h_1(x) + \ldots + w_k h_k(x). \quad (6)$$

Durch eine solche Kombination soll die Varianz des resultierenden Klassifikators reduziert und somit die Klassifikationsgüte erhöht werden [128]. Die unterschiedlichen Verfahren des Ensemble Learning unterscheiden sich hinsichtlich folgender Aspekte:

- der Menge an Trainingsinstanzen, die zur Bildung der einzelnen Funktionen $h_i, i \in \{1, \ldots, k\}$ zur Verfügung stehen,
- der Wahl der Gewichte $w_1, \ldots, w_k$,
- der zugrunde liegenden Klassifikationsmodelle für die Funktionen h_i. So können alle h_i ein gleiches Klassifikationsmodell (beispielsweise SVMs) nutzen oder es können unterschiedliche Klassifikationsmodelle genutzt werden.

Etablierte Techniken des Ensemble Learning sind Stacking, Bagging und Boosting. Beim *Stacking* werden unterschiedliche Klassifikationsmodelle benutzt, um die einzelnen Hypothesen h zu erzeugen. So können beispielsweise SVMs mit NB-Klassifikatoren kombiniert werden. Weiterhin stellt Stacking einen *Meta-Klassifikator* dar, da die einzelnen Gewichte $w_1, \ldots, w_k$ auf Basis der Trainingsdaten gelernt werden. Beim *Bagging* werden dagegen die gleichen Klassifikationsmodelle für die einzelnen Hypothesen verwendet. Zur Bildung der einzelnen Hypothesen $h_1, \ldots, h_k$ steht jeweils eine Teilmenge der verfügbaren Trainingsinstanzen zu Verfügung. Diese Teilmengen werden zufällig gewählt und müssen nicht disjunkt sein. Für die Gewichte der Hypothesen gilt bei Klassifikatoren mit numerischer Ausgabe $w_1 = \ldots = w_k = \frac{1}{k}$, somit haben alle Klassifikatoren den gleichen Einfluss auf die Klassifikationsentscheidung. Auch beim *Boosting* werden die gleichen Klassifikationsmodelle für die einzelnen Hypothesen verwendet. Boosting unterscheidet sich von Bagging darin, dass beim Boosting die Teilmengen zum Bilden der einzelnen Hypothesen nicht unabhängig gewählt werden. Die Bildung erfolgt iterativ. Zunächst werden für die Bildung des ersten Klassifikators h_1 zufällige Elemente aus der Trainingsmenge verwendet. Für die Bildung des zweiten Klassifikators h_2 werden die Trainingsinstanzen verwendet, die vom Klassifikator h_1 am schlechtesten klassifiziert wurden. Insgesamt gilt, dass für die Bildung des i-ten Klassifikators die Trainingsinstanzen verwendet werden, die von den auf den Hypothesen $h_1, \ldots, h_{i-1}$ aufbauenden Klassifikatoren am schlechtesten klassifiziert werden. Die Wahl der Gewichte $w_1, \ldots, w_k$ erfolgt beim Boosting in Abhängigkeit der Klassifikationsgüte der einzelnen Klassifikatoren. [174]

Zur Textklassifikation haben sich SVMs als bestes Klassifikationsmodell herausgestellt (siehe Abschnitt 2.2.2). Da durch diese Beobachtung von einer Kombination mit anderen Klassifikationsmodellen kein Mehrwert erwartet wird, wird vom Ansatz des Stacking, welches verschiedene Klassifikationsmodelle kombiniert, in dieser Arbeit abgesehen. Bei vergleichenden Untersuchungen zur Verwendung von Ensemble Learning in Verbindung mit SVMs zur Textklassifikation hat sich gezeigt, dass Bagging bessere Ergebnisse liefert als Boosting [41]. Weiterhin zeichnet sich Bagging durch eine einfachere Berechenbarkeit aus, da die einzelnen Klassifikatoren im Ensemble unabhängig voneinander und somit parallel trainiert werden können. Aus diesen beiden Gründen wird im Rahmen dieser Arbeit Bagging als Methode des Ensemble Learning mit dem Ziel eines robusten Klassifikators verwendet.

6.1.2 *Active Learning*

Settles et al. beschreiben *Active Learning* wie folgt [154, Seite 1]:

> *„The goal of active learning is to minimize the cost of training an accurate model by allowing the learner to choose which instances are labeled for training."*[1]

Da in dieser Arbeit Methoden zur domänenadaptiven Strukturierung von textuellen Dokumenten vorgestellt werden und die Identifikation von Meta-Attributen die Verwendung von Klassifikationsverfahren erfordert, bietet sich das Konzept des Active Learning zur Verwendung an: Durch den reduzierten Bedarf an Trainingsdaten und den damit verbundenen geringeren Aufwand zur Annotation im Vergleich zu anderen überwachten Lernverfahren, kann das Verfahren schneller und mit geringeren Kosten in einer neuen, unbekannten Domäne eingesetzt werden.

Die Herausforderung beim Active Learning besteht darin, dass die Instanzen, die annotiert werden sollen, identifiziert werden müssen. Dies sollten jene Instanzen sein, durch deren Verwendung als Trainingsinstanzen die größtmögliche Verbesserung des Klassifikators erzielt werden kann. Beim Active Learning muss zunächst eine kleine initiale Menge an Trainingsdokumenten zur Verfügung stehen, um ein initiales Klassifikationsmodell zu erstellen. Auf Basis dessen werden dann weitere zu annotierende Instanzen ausgewählt, welche zur Bildung eines neuen, idealerweise verbesserten Modells genutzt werden. Diese Schritte können iterativ wiederholt werden, indem jeweils das aktuelle Modell verwendet wird, um neue zu annotierende Instanzen zu identifizieren.

Für die Auswahl der zu annotierenden Instanzen existiert eine Vielzahl an Ansätzen. Fu et al. [52] klassifizieren diese in zwei Klassen:

1. Verwendung von Metriken, welche auf Basis der Klassifikationsungenauigkeit einzelner Instanzen berechnet werden und von der Unabhängigkeit der einzelnen Instanzen ausgehen; der Fokus der Ansätze dieser Klasse liegt auf dem Nutzen der Annotation der einzelnen Instanz,

2. Verwendung von Metriken, welche zusätzlich zur Klassifikationsungenauigkeit der einzelnen Instanzen Korrelationen zwischen Instanzen berücksichtigen; somit liegt der Fokus auf der Diversität und Repräsentativität der zu annotierenden Instanzen.

Durch Auswahl der Instanzen, von welchen der größte Nutzen durch deren Annotation erwartet wird, kann es bei den Methoden der ersten Klasse vorkommen, dass sehr ähnliche Instanzen ausgewählt werden und somit redundante Informationen in den nachträglich annotierten Instanzen auftauchen. Im Gegensatz dazu kann es bei den Methoden der zweiten Klasse durch den Fokus auf Diversität vorkommen, dass relevante Informationen verworfen werden. Dies ist gerade bei Clusterbildung der Instanzen problematisch, da mitunter die Randpunkte der Cluster, welche von hoher Relevanz für die Bestimmung der trennenden Grenze der SVM sind, vernachlässigt werden. Diese Problematik ist in Abbildung 8 dargestellt. Die Trenngrenze, die auf Basis der initialen Trainingsdaten erkannt wurde (Abbildung 8a) ist nicht

1 *„Das Ziel des Active Learning ist die Minimierung der Kosten für das Training eines genauen Modells durch die Wahl der zu labelnden Instanzen für das Training durch das Lernverfahren."* (freie Übersetzung durch den Autor)

optimal. Wenn nun die Instanzen, deren Abstand zur Trenngrenze minimal ist (Abbildung 8b), zur Nachannotation und zum iterativen Neutraining des Klassifikators verwendet werden, entsteht eine Trenngrenze, welche die erzeugende Funktion f sehr gut approximiert. Wenn wiederum nicht nur der Abstand zur Trenngrenze, sondern auch die Korrelation zwischen den Instanzen berücksichtigt wird (zweite Klasse, Abbildung 8c), dann wird der Clustermittelpunkt als repräsentative, zu annotierende Instanz verwendet statt des Randpunktes. Dies resultiert in einer weniger guten Approximation der erzeugenden Funktion f.

Im Rahmen dieser Arbeit werden Dokumente mit heterogenem Dokumentenlayout betrachtet (siehe Abschnitt 4.2), dies sorgt für eine starke Verteilung der Datenpunkte im Raum. Weiterhin gibt es Gruppen von Dokumenten aus gleichen Quellen und mit ähnlichem Layout. Dies trifft in der Domäne *Ausschreibungen studentischer Abschlussarbeiten* für die Ausschreibungen eines Fachgebietes (vergleiche Abschnitt 4.2.3) oder in der Domäne *Stellenanzeigen* auch für die Stellenausschreibungen eines Arbeitgebers (vergleiche Abschnitt 4.2.1) zu. Somit sind die Datenpunkte im Raum heterogen verteilt, es ist aber mit lokalen Clustern zu rechnen. Bei der Berücksichtigung der Korrelation würden eher die Zentroide der Cluster als zu annotierende Instanzen ausgewählt werden und nicht die Randpunkte der Cluster. Erstrebenswert ist jedoch gerade die Berücksichtigung dieser Randpunkte zur Identifikation der optimalen Trenngrenze. Weiterhin haben Verfahren der zweiten Klasse eine schlechtere Laufzeitkomplexität. Dies resultiert aus der benötigten Berechnung der Korrelationen zwischen den einzelnen Datenpunkten [52]. Aufgrund dieser Charakteristiken wird auf die erste Klasse der Active Learning Ansätze zurückgegriffen, welche zur Auswahl der zu annotierenden Instanzen keine Korrelationen zwischen den Instanzen berücksichtigen.

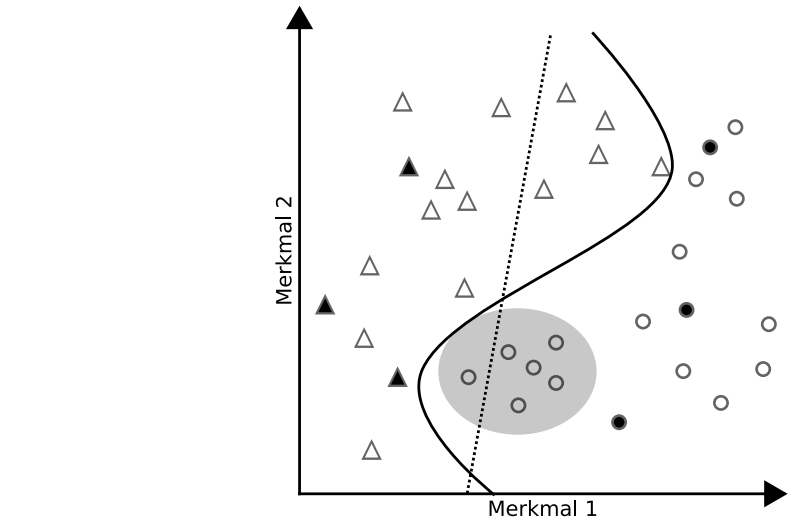
Die Klasse der Active Learning Verfahren, welche von der Unabhängigkeit der einzelnen Instanzen ausgehen, weist eine vergleichsweise geringere Laufzeitkomplexität auf. Jedoch sind auch hier Unterschiede hinsichtlich der Laufzeit zu erkennen. Am vorteilhaftesten schneiden hierbei die Verfahren des *Uncertainty Samplings* [86] ab, zu denen unter anderem das konfidenzbasierte Active Learning gehört.

Im konfidenzbasierten Active Learning [88] wird die Entscheidung, ob eine Annotation für eine Instanz x durchgeführt werden sollte, mittels einer *Query-Function* Q getroffen:

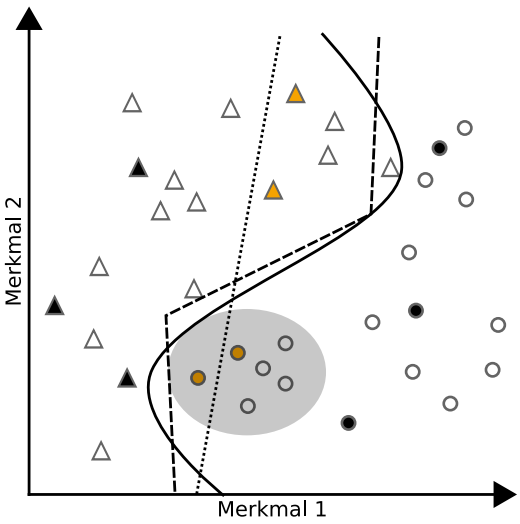
$$Q(x) = \begin{cases} 1 & \text{wenn } \mathtt{uncertainty}(x) \geqslant c \\ 0 & \text{sonst,} \end{cases} \tag{7}$$

wobei die Funktion $\mathtt{uncertainty}(x)$ die Unsicherheit des betrachteten Klassifikators hinsichtlich der Klassifikation von x beschreibt und c ein Grenzwert ist, welcher in Abhängigkeit der Anwendung zu wählen ist. Eine Instanz x sollte genau dann annotiert werden, wenn $Q(x)$ zu 1 ausgewertet wird. Außerdem kann ein weiterer Parameter k verwendet werden, der eine obere Grenze für die Anzahl der zu annotierenden Instanzen angibt. Wenn die Menge der Instanzen x, für die $Q(x) = 1$ gilt, größer als k ist, werden aus der Menge k zufällige Instanzen ausgewählt, die annotiert werden. Die verbleibenden Instanzen werden verworfen.

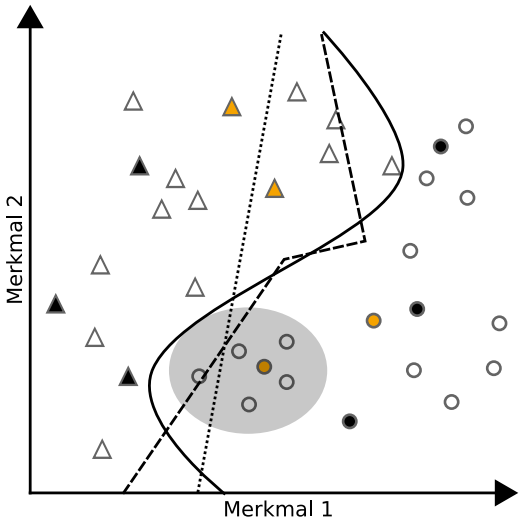
Bei der Verwendung von SVMs lässt sich die Konfidenz über den Abstand der klassifizierten Instanz von der separierenden Trennebene bestimmen. Das Inverse dieses Abstandes kann in normalisierter Form als $\mathtt{uncertainty}$-Funktion verwendet



(a) vor Nachannotation



(b) Nachannotation auf Basis der einzelnen, unabhängigen Instanzen und Verwendung dieser Instanzen zur Bildung einer neuen Funktion h



(c) Nachannotation unter Einbezug von Klassenkorrelationen und Verwendung dieser Instanzen zur Bildung einer neuen Funktion h

Abbildung 8: Beispielhafte Darstellung des Einflusses der Methodik zur Auswahl der zu annotierenden Instanzen, abgewandelt nach [52]; Dreiecke und Kreise repräsentieren die Instanzen der zwei Klassen, ausgefüllte Formen stellen Trainingsinstanzen dar, orangene Formen repräsentieren die zur Nachannotation ausgewählten Instanzen, die durchgezogene Linie stellt die Funktion f dar, die gepunktete Linie die aktuelle Hypothese h und die gestrichelte Linie in (b) und (c) die aktualisierte Hypothese, die graue Ellipse hebt einen Cluster hervor

werden, da ein hoher Abstand eine hohe Sicherheit der Klassenzugehörigkeit und somit einen niedrigen Wert für die $uncertainty$-Funktion repräsentiert.

6.1.3 *Diskussion der Grundlagen*

In diesem Abschnitt wurde ein Überblick über verschiedene Methoden im Bereich des Ensemble Learning und des Active Learning zur Textklassifikation gegeben. Insbesondere wurde dabei untersucht, welche der Methoden in diesen Bereichen für den in dieser Arbeit adressierten Anwendungszweck zur Identifikation von Meta-Attributen geeignet sind. Dabei hat sich Bagging als Ensemble Learning Technik unter der Verwendung von SVMs aufgrund der hohen Klassifikationsgüte und der geringen Komplexität als geeignet herausgestellt. Weiterhin wurde gezeigt, dass konfidenzbasiertes Active Learning geeignet ist um mit geringer Komplexität eine höhere Domänenadaptivität der Identifikation von Meta-Attributen zu erreichen. Die Kombination von Bagging mit SVMs und konfidenzbasiertem Active Learning wurde beispielsweise von Li und Snoek [89] vorgestellt. Im Fokus steht hierbei jedoch keine Textklassifikation, sondern die Vorhersage der Relevanz von Tags für Bilder. Insbesondere wird in diesem Ansatz stark zwischen negativen und positiven Instanzen unterschieden. Insgesamt wird ein Ensemble von Klassifikatoren häufiger zur Identifikation der zu annotierenden Instanzen verwendet, jedoch muss in den existierenden Ansätzen das komplette Ensemble jeweils neu trainiert werden, was zu einer hohen Laufzeitkomplexität führt [52]. Aus diesem Grund wird in dieser Arbeit ein kombiniertes Verfahren vorgestellt, bei dem das Ensemble nicht neu trainiert werden muss, aber trotzdem die Vorteile des Ensemble Learning und des Active Learning genutzt werden sollen.

6.2 BESCHREIBUNG DES VERFAHRENS

Eine Übersicht über die Komponenten des *Combined Ensemble and Fast Active Learner (CENFA)*-Systems und ihr Zusammenspiel ist in Abbildung 9 dargestellt. Im Folgenden werden zunächst die Komponenten des Systems vorgestellt und dann die unterschiedlichen Phasen des Trainings- und Klassifikationsprozesses beschrieben.

6.2.1 *Komponenten*

Das System besteht aus zwei wesentlichen Komponenten: dem *Basisklassifikator* und dem *Spezialklassifikator*. Beides sind unabhängige Klassifikatoren, welche sich jedoch durch ihre Eigenschaften unterscheiden. Durch ihre Verbindung zu einem Gesamtsystem sollen die positiven Eigenschaften beider Klassifikatoren ausgenutzt werden, um ein robustes, aber gleichzeitig domänenadaptives System zu entwickeln (vergleiche Ziele der Dissertation in Abschnitt 1.2).

6.2.1.1 *Basisklassifikator*

Der *Basisklassifikator* besteht intern aus mehreren Klassifikatoren, welche per Bagging zu einem Ensemble zusammengesetzt sind. Aufgrund der guten Ergebnisse von SVMs zur Textklassifikation (vergleiche Abschnitt 2.2.2) werden diese im Basisklas-

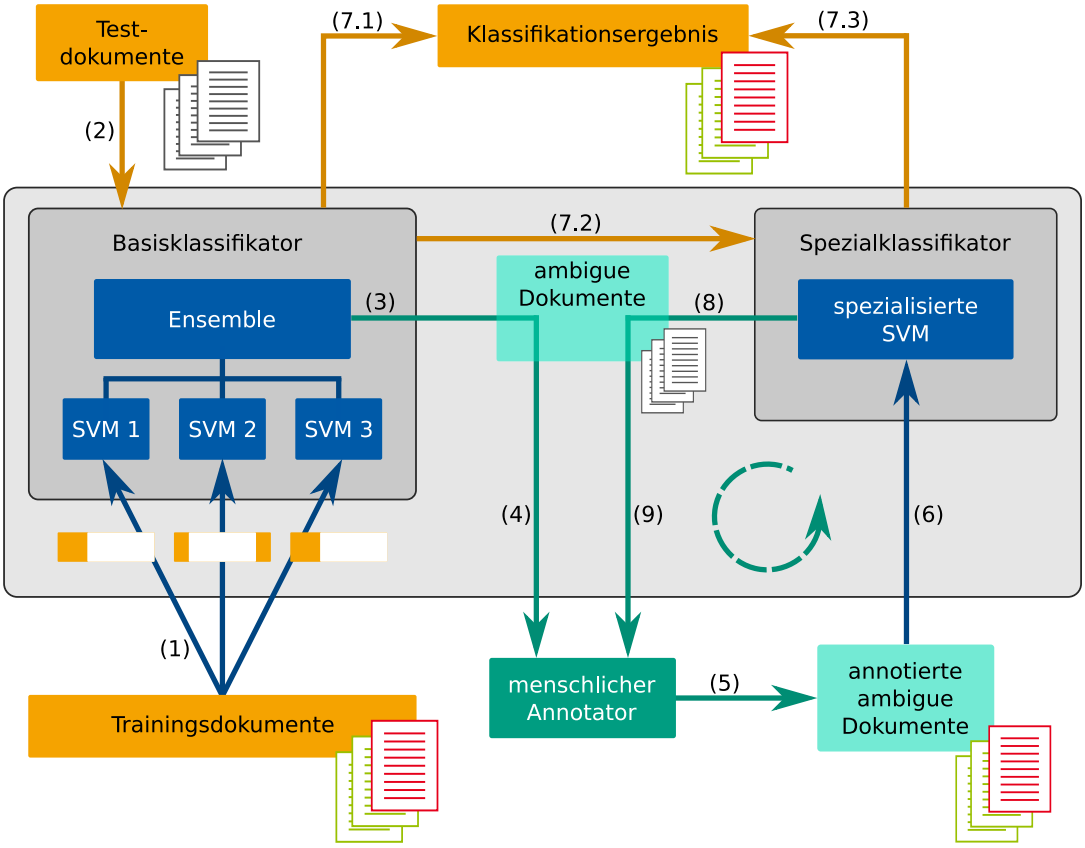


Abbildung 9: Darstellung der CENFA-Architektur; die angegebenen Nummern definieren die Ablaufreihenfolgen zum Training des Systems und zur anschließenden Klassifikation durch das System; dunkelblaue Pfeile symbolisieren das Training des Systems, orangene Pfeile den Klassifikationsprozess, türkise Pfeile die Sammlung und Annotation ambiguer Dokumente; die Farben rot und grün der Dokumente symbolisieren beispielhafte Klassen, bei grau dargestellten Dokumenten ist die Klasse nicht bekannt

sifikator eingesetzt. Das Konzept würde jedoch auch die Verwendung anderer Klassifikatoren erlauben. Der Konfidenzwert des Ensembles ist definiert durch das arithmetische Mittel der Konfidenzwerte der einzelnen SVMs. Diese sind jeweils durch den Abstand der zu klassifizierenden Instanz von der separierenden Hyperebene der SVM definiert. Das Vorzeichen dieses Wertes definiert dabei das Klassifikationsergebnis. Der Basisklassifikator wird initial trainiert und verändert sich während der Nutzung des Systems nicht. Die Verwendung eines Ensembles zielt auf die hohe Robustheit des Basisklassifikators ab. Das Training des Basisklassifikators ist relativ zeitintensiv, da die SVMs im Ensemble einzeln trainiert werden müssen. Dies sorgt für eine hohe Laufzeit für das Neutraining des Basisklassifikators, weshalb zusätzlich der Spezialklassifikator eingeführt wird.

6.2.1.2 *Spezialklassifikator*

Der *Spezialklassifikator* besteht im Gegensatz dazu aus einem einzelnen Klassifikator. Zum Einsatz kommt aus den selben Gründen wie beim Basisklassifikator eine SVM, wobei auch die Verwendung anderer Klassifikatoren denkbar wäre. Die Verwendung eines solchen einzelnen Klassifikators erlaubt das regelmäßige Neutrainieren des

Spezialklassifikators mit geringem Zeitaufwand. Hierzu wird das Konzept des Active Learning eingesetzt, wobei die Auswahl der nachzuannotierenden Instanzen unter Verwendung des Konfidenzwerts des Basisklassifikators per konfidenzbasiertem Active Learning durchgeführt wird.

6.2.2 *Phasen des Trainings und der Klassifikation*

Während bei klassischen überwachten Lernverfahren eine strikte Trennung zwischen Training und Nutzung (Testen) eines Klassifikators herrscht, ist diese Grenze bei Verfahren des Active Learning weniger strikt. Dort liegen klassischerweise ebenso zwei Phasen vor: eine erste Phase, in der nur trainiert wird, und eine zweite Phase, in der klassifiziert (getestet) und iterativ weiterhin trainiert wird. Beim CENFA-Verfahren liegen jedoch drei Phasen vor, welche im Folgenden beschrieben werden. Die verwendeten Nummern beziehen sich auf die Nummern in Abbildung 9, die die einzelnen Schritte repräsentieren.

6.2.2.1 *Setup-Phase 1*

Zunächst werden die initial zur Verfügung stehenden Trainingsdokumente verwendet, um den Basisklassifikator zu trainieren (1). Hierbei werden die einzelnen SVMs des Ensembles mit unterschiedlichen Teilmengen des Trainingsdatensatzes trainiert. Sobald dieses Training erfolgt ist, ist das System in der Lage, Klassifikationen per Bagging vorzunehmen (2). Durch die ersten dieser Klassifikationsentscheidungen wird die zweite Setup-Phase initiiert.

6.2.2.2 *Setup-Phase 2*

Für zu klassifizierende Dokumente (2) kann der nun trainierte Basisklassifikator eine Klassifikationsentscheidung treffen (7.1). Weiterhin wird der durch den Basisklassifikator bestimmte Konfidenzwert zur Bestimmung ambiguer Dokumente verwendet. Liegt der Konfidenzwert unter dem *Annotations-Konfidenz-Grenzwert (AKG)*, so wird das zu klassifizierende Dokument als ambigue gekennzeichnet (3). Diese als ambigue markierten Dokumente werden anschließend einem menschlichen Annotator vorgelegt (4), da die Klassenlabel unbekannt sind. Zusammen mit den vom Annotator vergebenen Klassenlabeln entsteht so eine weitere Menge annotierter Trainingsdaten (5). Sobald eine ausreichende Menge annotierter, ambiguer Dokumente vorliegt, werden diese wiederum verwendet um den Spezialklassifikator initial zu trainieren (6).

6.2.2.3 *Regulärer Betrieb*

Im regulären Betrieb sind sowohl der Basisklassifikator als auch der Spezialklassifikator trainiert. Wenn nun weitere Klassifikationsaufgaben zur Verfügung stehen (2), werden diese weiterhin zunächst an den Basisklassifikator geleitet. Wenn dessen Konfidenzwert über dem AKG liegt, wird das Klassifikationsergebnis ausgegeben (7.1). Falls der Konfidenzwert jedoch kleiner als der AKG ist, wird das Dokument an den Spezialklassifikator weitergereicht (7.2), welcher dann eine Entscheidung trifft (7.3). Parallel dazu werden weiterhin ambigue Dokumente gesammelt. Dies sind im

regulären Betrieb jedoch die Dokumente, für die der Konfidenzwert des Spezialklassifikators unter dem Grenzwert AKG liegt (8). Auch diese Dokumente werden an den menschlichen Annotator weitergeleitet (9), annotiert (5) und zum Neutraining des Spezialklassifikators genutzt (6). Dieser sich wiederholende Zyklus soll einer stetigen Verbesserung des Spezialklassifikators dienen.

6.3 EVALUATION DES VERFAHRENS

Um Aussagen über die Güte des Verfahrens treffen zu können, wurde das Verfahren systematisch in der Anwendungsdomäne *Stellenanzeigen* evaluiert. Im Folgenden wird zunächst auf die dafür verwendeten Evaluationsdaten eingegangen, anschließend wird die genutzte Konfiguration für die Evaluation erläutert und zuletzt werden die Ergebnisse der Evaluation präsentiert.

6.3.1 *Evaluationsdaten*

Zur Evaluation wurde ein Korpus bestehend aus 10.300 deutschsprachigen Dokumenten aus der Domäne *Stellenanzeigen* verwendet, wobei die einzelnen Dokumente jeweils ein Stellenangebot enthalten. Dieser Korpus dient als Goldstandard zur Evaluation. Die Dokumente wurden von einem Projektpartner zur Verfügung gestellt und stammen von Webseiten unterschiedlichen Ursprungs, wie beispielsweise Unternehmenswebseiten, Stellenportalen oder Zeitungswebseiten. Jegliche Auszeichnungselemente, wie beispielsweise HTML-Tags, wurden aus den Dokumenten entfernt. Die Dokumente sind annotiert mit einem oder mehreren Klassenlabeln, welche das Berufsfeld widerspiegeln, in dem die zu vergebene Stelle zu verorten ist. Dieses Label stellt ein Meta-Attribut für das Dokument der Stellenanzeige dar, die Klassen repräsentieren die Attributwerte. Insgesamt wurde ein Pool von 103 verschiedenen Klassenlabeln verwendet, wobei ein Dokument im Durchschnitt mit 4,25 Labeln ($\sigma = 1,86$) annotiert wurde. Im Rahmen der Evaluation findet eine Fokussierung auf die fünf am häufigsten verwendeten Klassen statt. Eine Übersicht über die Klassen und ihre jeweiligen Auftrittshäufigkeiten ist Tabelle 17 zu entnehmen. Alle Instanzen, die für eine dieser fünf spezifischen Klassen kein Label haben, werden als nicht zur Klasse gehörig angenommen und können somit als negatives Trainingsbeispiel für den Klassifikator genutzt werden.

Tabelle 17: Betrachtete Klassen im Evaluationskorpus zusammen mit der Anzahl positiv annotierter Instanzen pro Klasse

Abkürzung	Klasse	Anzahl
SE	Softwareentwicklung	2.077
TM	Technisches Management	1.727
V	Vertrieb	1.587
PQS	Produktions- und Qualitätssicherheit	1.501
TED	Technische Entwicklung und Design	1.069

6.3.2 *Evaluationsmethodik*

Da im vorgestellten Evaluationskorpus Dokumente mit mehr als einem Label versehen sind, liegt ein Multi-Label-Klassifikationsproblem vor (vergleiche Abschnitt 2.2.2). Zur Lösung eines solchen Problems ist ein Multi-Label-Klassifikator notwendig. Der vorgestellte CENFA-Ansatz ist jedoch für binäre Klassifikationsentscheidungen entwickelt. Aufgrund der Unabhängigkeit der einzelnen Klassenlabel kann das Multi-Label-Problem jedoch auf mehrere Single-Label-Probleme reduziert werden [97]. Bei dieser Reduktion wird für jedes Label ein eigener CENFA-Klassifikator trainiert und die Entscheidungen der einzelnen Klassifikatoren werden unabhängig voneinander getroffen werden.

Das vorgestellte CENFA-Verfahren erfordert interaktives Feedback durch menschliche Annotatoren zum Annotieren der als ambigue identifizierten Dokumente. Um dies zu simulieren, wird im Rahmen der Evaluation eine Teilmenge der Evaluationsdaten genutzt. Diese Teilmenge wird dem System zur Klassifikation vorgelegt, wobei die vorab bestimmten Label dem System nicht zur Verfügung stehen. Alle Dokumente, die das System als ambigue annotiert, werden zusammen mit ihren Labeln der Menge der annotierten, ambiguen Dokumente hinzugefügt. Dieser Schritt spiegelt die Annotation durch den Menschen in der Evaluation wider.

Hieraus ergibt sich für die Evaluation eine Aufteilung der Evaluationsdaten E in drei disjunkte Mengen, wobei $E_{train} \cup E_{simulate} \cup E_{test} = E$ gilt:

- Die Dokumente in E_{train} werden zusammen mit ihren Klassenlabeln verwendet, um den Basisklassifikator zu trainieren.
- Alle Dokumente in $E_{simulate}$ werden vom Basisklassifikator klassifiziert. Die Elemente, die hierbei als ambigue identifiziert werden, bilden die Menge $E_{ambigue}$. Die Dokumente in dieser Menge werden zusammen mit ihren Klassenlabeln als Trainingsdaten an den Spezialklassifikator weitergeleitet.
- Die Dokumente in E_{test} werden als Testdaten genutzt, um unter Verwendung der Klassenlabel die Güte des CENFA-Systems zu evaluieren.

Um Aussagen über das Verhalten des Verfahrens bei unterschiedlichen Mengen an Evaluationsdaten treffen zu können, wird mit unterschiedlich großen Teilmengen der verfügbaren Evaluationsdaten als Menge E gearbeitet. Es findet eine Reduktion auf 10%, 25%, 50% und 75% der verfügbaren Evaluationsdaten statt. Für die jeweilige Reduktion wird eine zufällige Teilmenge entnommen. Insbesondere von Interesse ist dabei die damit verbundene Variierung der Kardinalitäten der Mengen E_{train} und $E_{simulate}$, da von diesen, durch ihre Verwendung als Trainingsdaten[2], ein wesentlicher Einfluss auf die Klassifikationsgüte erwartet wird (vergleiche die in Abschnitt 2.2.1 gegebenen Erläuterungen zur Lernkurve eines Klassifikators). Abbildung 10 gibt einen Überblick über die Zusammenhänge zwischen den einzelnen zuvor beschriebenen Mengen.

Die Aufteilung der Menge E in die genannten Teilmengen erfolgt zufällig. Um den Effekt unrepräsentativer Aufteilungen zu vermeiden, wird diese Aufteilung 10-fach

2 Aus der Menge $E_{simulate}$ werden nur die Elemente in der Teilmenge $E_{ambigue}$ zum Training genutzt. Die Größe dieser Teilmenge ist jedoch nicht gezielt durch Reduktion des Evaluationskorpus beeinflussbar.

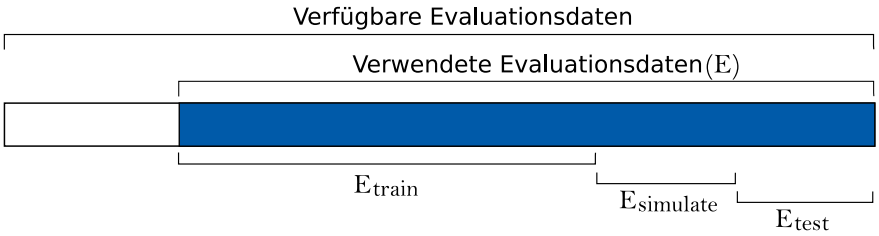


Abbildung 10: Darstellung der Mengen zur Evaluation auf Basis der Gesamtmenge der verfügbaren Evaluationsdaten

stratifiziert wiederholt und die unten präsentierten Ergebnisse stellen das arithmetische Mittel für die Evaluationsläufe mit den unterschiedlichen Aufteilungen dar. Dies stellt eine abgewandelte Form der 10-fach stratifizierten Kreuzvalidierung dar, da nicht wie klassischerweise in zwei Teilmengen[3] unterteilt wird, sondern in drei Teilmengen (vergleiche Abschnitt 2.2.3.3). Für die Evaluation wird die Aufteilung in unterschiedlich große Teilmengen vorgenommen, um den Einfluss der Anzahl der Dokumente in E_{train} und $E_{simulate}$ auf die Klassifikationsgüte untersuchen zu können.

Um die Güte des Systems zu bewerten, werden drei Abwandlungen des zuvor vorgestellten CENFA-Verfahrens mit dem regulären CENFA-Verfahren verglichen. Als Gütekriterien werden die Klassifikationsgenauigkeit in Form der Accuracy zur Bewertung der Effektivität und die Zeit zum Trainieren des Klassifikators zur Bewertung der Effizienz des Klassifikators verwendet. Die drei Varianten sind im Folgenden beschrieben.

Die Architektur des *Random*-Klassifikators ist identisch zum CENFA-Klassifikator. Jedoch wird zum Training des Spezialklassifikators nicht die Menge $E_{ambigue}$, sondern eine zufällige Teilmenge E_{random} aus $E_{simulate}$ verwendet. Mittels eines Vergleichs mit dem CENFA-Klassifikator soll untersucht werden, inwiefern die gezielte Auswahl der Dokumente aus $E_{simulate}$ durch das Active Learning Konzept zum Training des Spezialklassifikators einen Einfluss auf die Klassifikationsgüte hat. Um vergleichende Aussagen treffen zu können, gilt $|E_{random}| = |E_{ambigue}|$, so dass beide Klassifikatoren mit der gleichen Anzahl an Instanzen trainiert werden, da eine variierende Anzahl Einfluss auf die Güte haben könnte.

Der *Extended*-Klassifikator besteht allein aus einem Ensemble von Klassifikatoren, analog zum Basisklassifikator des CENFA-Ansatzes. Sobald die Menge $E_{ambigue}$ vorliegt, wird das komplette Ensemble mit der Menge $E_{ambigue} \cup E_{train}$ neu trainiert. Hierbei soll untersucht werden, wie sich Klassifikationsgüte und Zeitanforderungen für das erneute Training im Vergleich zum CENFA verhalten.

Als drittes Vergleichskonzept wird eine einzelne SVM verwendet, welche mit der Menge $E_{random} \cup E_{train}$ trainiert wird. Dieses Konzept wird im Folgenden als *Random Single Support Vector Machine (RSSVM)* bezeichnet. Durch den Vergleich mit diesem Ansatz soll untersucht werden, welchen Effekt die Verwendung eines Ensembles hinsichtlich der Klassifikationsgüte hat. Weiterhin soll das Zeitverhalten insbesondere für die initiale Trainingsphase analysiert werden.

Es ist festzuhalten, dass die drei genannten Konzepte sich hinsichtlich ihrer Architektur und hinsichtlich der zum Training verwendeten Instanzen unterscheiden. Die Anzahl der insgesamt zum Training der Konzepte zur Verfügung stehenden annotierten Trainingsdaten ist konstant. Als Testmenge wird durchgehend E_{test} verwen-

3 Trainingsdaten und Testdaten

Tabelle 18: Übersicht über zur Evaluation verwendete Klassifikatoren, jeweilige Trainingsmengen für den Basis- und den Spezialklassifikator, sowie Fokus der vergleichenden Evaluationen

	Trainingsmenge		
Klassifikator	Basisklassifikator	Spezialklassifikator	Fokus der vergleichenden Evaluation
CENFA	E_{train}	$E_{ambigue}$	vorgestellter Ansatz
Random	E_{train}	E_{random}	Klassifikationsgüte bei der Benutzung zufälliger Instanzen statt ambiguer Instanzen
Extended	$E_{ambigue} \cup E_{train}$	-	Klassifikationsgüte/Zeitverhalten mit komplettem Neutraining
RSSVM	-	$E_{random} \cup E_{train}$	Klassifikationsgüte/Zeitverhalten ohne Ensemble

det. Tabelle 18 gibt einen zusammenfassenden Überblick über die Klassifikatoren, die unterschiedlichen Trainingsmengen und den Fokus der jeweiligen Evaluation.

Da Klassifikatoren, wie die verwendeten SVMs, nur numerische Repräsentationen der zu klassifizierenden Instanzen klassifizieren können, müssen die Dokumente des Evaluationskorpus vorverarbeitet werden. Hierzu wird zunächst unter der Verwendung eines deutschen Tokenizers eine Tokenisierung vorgenommen. Die insgesamt 10.000 häufigst vorkommenden Token (Unigramme) werden als Grundlage zur Vektorbildung verwendet, wobei eine Gewichtung unter Verwendung des TF-IDF-Maßes vorgenommen wird (siehe Abschnitt 2.2.2).

6.3.3 *Parameter zur Evaluation*

Um den Effekt der unterschiedlichen Trainingsmengen für Basisklassifikator und Spezialklassifikator zu untersuchen, wird der *Aufteilungsfaktor (AF)* als Parameter eingeführt. Dieser gibt an, mit welchem Verhältnis die Gesamtmenge E in ihre Teilmengen aufgeteilt wird. Der AF beschreibt den Anteil der Dokumente, die sich in E_{train} befinden, also $\frac{|E_{train}|}{|E|}$. Die verbleibenden Elemente werden gleichmäßig auf $E_{simulate}$ und E_{test} aufgeteilt, also $|E_{simulate}| = |E_{test}|$. Somit bedeutet beispielsweise $AF = 0,75$, dass E_{train} 75% des Evaluationskorpus E enthält, wohingegen $E_{simulate}$ und E_{test} jeweils 12,5% des Evaluationskorpus enthalten. Ein hoher Wert für AF bedeutet also eine große Anzahl an Trainigsdaten für den Basisklassifikator und eine kleinere Anzahl an potentiellen Trainingsdaten für den Spezialklassifikator.

Weiterhin wird der Konfidenzwert AKG variiert, um dessen Einfluss auf die Güte des Systems zu untersuchen. Wenn dieser niedrig gewählt wird, stehen wenige Dokumente zum Training des Spezialklassifikators zu Verfügung, da aus der Menge $E_{simulate}$ nur ein kleiner Anteil in die Menge $E_{ambigue}$ übernommen wird. Außerdem werden bei einem niedrigen Wert AKG auch nur wenige Klassifikationsentscheidungen vom Spezialklassifikator getroffen. Bei einem hohen Wert wiederum werden

mehr Dokumente als ambigue notiert und stehen zum Training des Spezialklassifikators zur Verfügung.

Die Parameter AF und AKG haben große wechselseitige Abhängigkeiten, da beide einen Einfluss auf die Anzahl an Dokumenten haben, die für das Training des Spezialklassifikators zur Verfügung stehen. Es wird erwartet, dass diese Anzahl einen maßgeblichen Einfluss auf die Klassifikationsgüte hat. Aufgrund dieser Abhängigkeiten werden diese beiden Parameter gemeinsam evaluiert. Der Fokus der Evaluation liegt auf der Untersuchung der Wahl des AKG, da nur dieser in der Nutzung des Verfahrens beeinflusst werden kann. Der AF dient ausschließlich zu Evaluationszwecken.

Der dritte und letzte zu variierende Parameter ist die Anzahl der einzelnen Klassifikatoren im Ensemble des Basisklassifikators. Bei einem hohen Wert für diesen Parameter stehen für die einzelnen Klassifikatoren des Spezialklassifikators weniger Dokumente zur Verfügung, da die vorhandenen Elemente in E_{train} auf die Klassifikatoren aufgeteilt werden. Bei einem niedrigen Wert könnte jedoch der angestrebte Vorteil der Robustheit des Ensembles abnehmen, da weniger Klassifikatoren zum Treffen einer Mehrheitsentscheidung zur Verfügung stehen. In einer Untersuchung von Breiman [22] hat sich gezeigt, dass schon ab 25 Klassifikatoren keine wesentliche Verbesserung mehr erzielt werden kann, daher wird im Rahmen dieser Dissertation die Verwendung von maximal 40 Klassifikatoren im Ensemble untersucht.

6.3.4 *Implementierung*

Das beschriebene CENFA-Verfahren, die Vergleichsverfahren und das Evaluationskonzept wurden unter Verwendung von Java umgesetzt. Weiterhin wurde das Framework zum maschinellen Lernen Weka [63] in Version 3.6 genutzt. Als SVM-Lösungsverfahren wurde auf den SMO [78, 127] Algorithmus, welcher in der Weka-Implementierung vorliegt (`weka.classifiers.functions.SMO`), zurückgegriffen. Die Parameter der SMO-Implementierung wurden bei ihren Standardwerten belassen. Variationen dieser Parameter werden nicht evaluiert, da dies nicht im Fokus der Arbeit steht. Zur Tokenisierung der Dokumente wurde die Klasse `weka.core.tokenizers.AlphabeticTokenizer` erweitert, so dass auch bei deutschen Umlauten eine korrekte Tokenisierung stattfindet.

6.3.5 *Ergebnisse*

In diesem Abschnitt werden die Ergebnisse der Evaluationen vorgestellt. Dargestellt sind jeweils die Ergebnisse nach einer Iteration, in der alle Instanzen aus $E_{simulate}$ klassifiziert wurden und die ambiguen dieser Instanzen als Trainingsinstanzen für den Spezialklassifikator verwendet werden. Dabei wird zunächst auf die Ergebnisse von zwei vorab stattgefundenen Evaluationsserien eingegangen, bei denen die beste Wahl der zu setzenden Parameter AF und AKG sowie die Zahl der SVMs im Basisklassifikator bestimmt wird. Im Folgenden wird die Klassifikationsgüte des CENFA-Verfahrens für unterschiedliche Mengen an Evaluationsdaten vorgestellt. Weiterhin wird gezeigt, wie sich der CENFA-Ansatz für zwei verschiedene Korpusgrößen im Vergleich zu den verwandten Ansätzen verhält. Dabei wird neben der Klassifikationsgüte ein Augenmerk auf die Laufzeit der Verfahren gelegt.

6.3.5.1 *Bestimmung des Annotationskonfidenzgrenzwerts und des Aufteilungsfaktors*

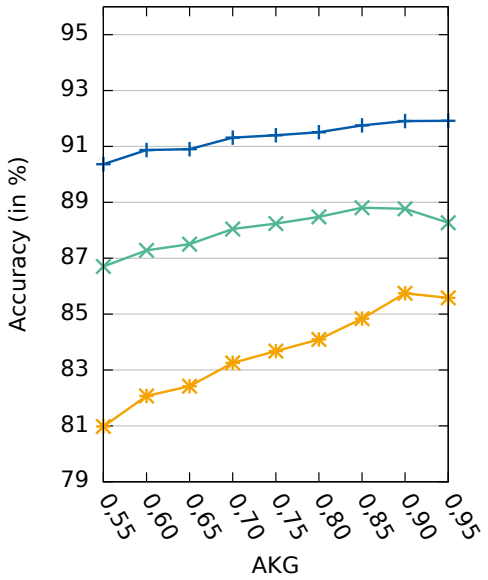
Da, wie zuvor beschrieben, davon auszugehen ist, dass die Parameter AKG und AF maßgeblichen Einfluss auf die Klassifikationsgüte haben, wird in diesem Abschnitt untersucht, wie sich die Wahl dieser Parameter auswirkt. Bei ersten Analysen hat sich gezeigt, dass der Einfluss dieser Parameter relativ unabhängig von der betrachteten Klasse ist. Daher werden im Folgenden die Ergebnisse für die Klassifikation in die Klasse *SE* präsentiert.

Abbildung 11 zeigt den Einfluss des AKG auf die Accuracy bei unterschiedlicher Wahl des AF. Die dargestellten Werte für AF wurden so gewählt, dass der Einfluss deutlich zu identifizieren ist. Bei einem AF von 0,2 wird ein großer Anteil der Evaluationsdaten für das Training des Spezialklassifikators verwendet, wohingegen bei $AF = 0,9$ nur sehr wenige Dokumente für das Training des Spezialklassifikators zur Verfügung stehen. Die Wahl $AF = 0,5$ repräsentiert einen Mittelwert zwischen diesen beiden Extremwerten. Dargestellt ist der Verlauf bei Verwendung von 10%, 25% und 50% der verfügbaren Evaluationsdaten. Bei Verwendung von 75% oder 100% Evaluationsdaten sind die im Folgenden beschriebenen Effekte nicht so deutlich erkennbar, da genügend Trainingsdokumente für das Training des Spezialklassifikators und des Basisklassifikators zur Verfügung stehen.

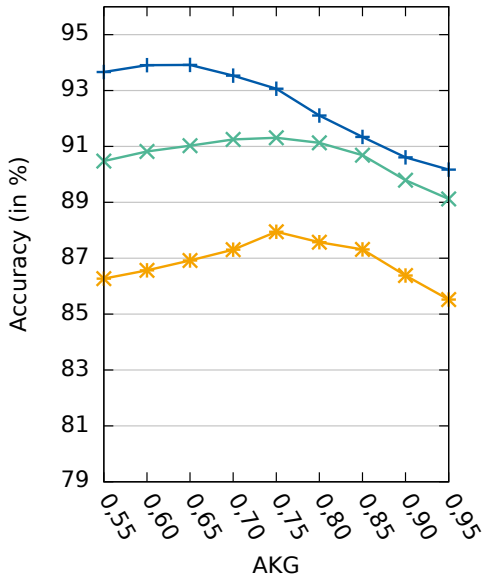
Bei Betrachtung der Graphen fällt auf, dass der allgemeine Trend stark von der Wahl des AF abhängt. Während bei $AF = 0,2$ (Abbildung 11a) eine Steigung der Klassifikationsgüte bei steigendem AKG zu beobachten ist, fällt bei $AF = 0,9$ (Abbildung 11c) die Klassifikationsgüte bei steigendem AKG. Bei der mittleren Wahl $AF = 0,5$ (Abbildung 11b) ist wiederum eine zunächst steigende und dann fallende Klassifikationsgüte zu beobachten. Die maximale Klassifikationsgüte ist bei $AF = 0,2$ deutlich schlechter als bei $AF = 0,5$ und $AF = 0,9$.

Bei $AF = 0,2$ enthält $E_{simulate}$ relativ viele Dokumente und für das Training des Basisklassifikators stehen nur wenige Trainingsdokumente zur Verfügung. Bei einem hohen AKG werden viele Dokumente als ambigue gekennzeichnet und stehen für das Training des Spezialklassifikators zur Verfügung. Da der Basisklassifikator nur mit wenigen Dokumenten trainiert ist, kommt es häufig zu Fehlklassifizierungen durch diesen. Dieser Effekt hat insbesondere bei niedrigem AKG Auswirkungen, da in diesem Fall die meisten Klassifikationsentscheidungen vom Basisklassifikator getroffen werden. Die große Zahl der Dokumente, die für das Training des Spezialklassifikators bei hohem AKG zur Verfügung stehen, ermöglicht ein gutes Training des Spezialklassifikators; gleichzeitig werden die meisten Klassifikationsentscheidungen vom Spezialklassifikator getroffen, was einen positiven Einfluss auf die Klassifikationsgüte hat.

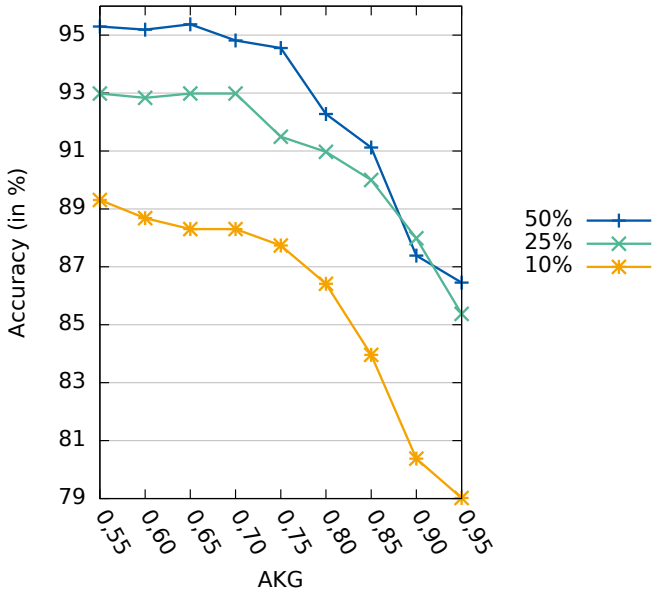
Bei dem anderen Extremwert $AF = 0,9$ enthält $E_{simulate}$ relativ wenige Dokumente. E_{train} dagegen enthält relativ viele Dokumente. Dies ermöglicht ein gutes Training des Basisklassifikator, während dem Spezialklassifikator nur wenige Trainingsdokumente zur Verfügung stehen. Bei einem geringen AKG überwiegt die gute Klassifikationsgüte des Basisklassifikators. Auch bei einer leichten Erhöhung des AKG (bis etwa 0,7) kann diese Klassifikationsgüte gehalten werden. Bei einem höheren AKG werden viele Entscheidungen an den Spezialklassifikator weitergeleitet, dieser führt jedoch häufiger Fehlklassifikationen durch, was eine Verschlechterung der Klassifikationsgüte des CENFA zur Folge hat.



(a) AF = 0,2; Anzahl der SVM = 10



(b) AF = 0,5; Anzahl der SVM = 10



(c) AF = 0,9; Anzahl der SVM = 10

Abbildung 11: Einfluss des Parameters AKG auf die Klassifikationsgüte bei Nutzung verschiedener Anteile des Evaluationskorpus und in Abhängigkeit der Wahl des Parameters AF

Bei einer mittleren Wahl für AF (0,5) sind beide Effekte zu erkennen. Bei niedrigem AKG ist die Klassifikationsgüte nicht bei ihrem Maximum. Hier werden die meisten Klassifikationsentscheidungen vom Basisklassifikator getroffen, welcher jedoch nicht vollständig trainiert ist. Dies ist insbesondere zu erkennen, wenn nur geringe Prozentsätze der Evaluationsdaten genutzt werden (10% beziehungsweise 25%). Bei steigendem AKG wird die maximale Klassifikationsgüte erreicht. Bei hohem AKG sinkt die Klassifikationsgüte jedoch wieder ab, da viele Klassifikationsentscheidungen von dem nicht vollständig trainierten Spezialklassifikator getroffen werden. Insgesamt wird die beste Klassifikationsgüte also erreicht, wenn sowohl für den Basisklassifikator als auch für den Spezialklassifikator eine ausreichend hohe Zahl an Trainingsdokumenten zur Verfügung stehen.

Tabelle 19 zeigt die Werte für AF und AKG bei denen bei unterschiedlichen Anteilen der Evaluationsdaten die höchste Klassifikationsgüte erreicht wurde. Insgesamt sollte der AF also zwischen 0,65 und 0,8 gesetzt werden und der AKG zwischen 0,7 und 0,8. Um Vergleichbarkeit über die unterschiedlichen Auswertungen zu ermöglichen, wird für die im Folgenden präsentierten Experimente sowohl für AF als auch für AKG ein konstanter Wert von 0,7 verwendet.

Tabelle 19: Übersicht über die Konfigurationen von AF und AKG, mit denen in Abhängigkeit des Anteils der verwendeten Evaluationsdaten die höchste Klassifikationsgüte erzielt wurde

Anteil der Evaluationsdaten	AF	AKG
10%	0,80	0,70
25%	0,75	0,70
50%	0,75	0,70
75%	0,70	0,75
100%	0,65	0,80

6.3.5.2 *Bestimmung der Zahl der Klassifikatoren im Ensemble*

Als weiterer Parameter zur direkten Beeinflussung der Klassifikationsgüte wird in diesem Abschnitt die Anzahl der SVMs im Ensemble des Basisklassifikators untersucht. Für die Verwendung von 10% und 25% der Evaluationsdaten (Abbildung 12 beziehungsweise Abbildung 13) wurde eine getrennte Darstellung für die einzelnen Klassen gewählt. Dahingegen werden bei der Verwendung von 50%, 75% und 100% der Evaluationsdaten (Abbildung 14) repräsentativ die Ergebnisse für die Klasse *SE* dargestellt, da festgestellt wurde, dass die Klassifikationsgüte sich hier für die anderen Klassen sehr ähnlich verhält. Die konkreten Werte der Klassifikationsgüte variieren zwar zwischen den Klassen, aber die Veränderung der Zahl der SVMs hat über die einzelnen Klassen hinweg sehr ähnliche Auswirkungen.

Bei Verwendung von 10% der Evaluationsdaten (Abbildung 12) ist über alle Klassen hinweg bei steigender Zahl der SVM zunächst eine Steigerung der Klassifikationsgüte zu erkennen. Ab etwa 15 SVMs variiert jedoch die Auswirkung der Hinzunahme weiterer SVMs. Während bei den Klassen *TM* und *PQS* eine stetige Steige-

rung bis hin zu 40 SVMs zu beobachten ist, ist für die anderen Klassen eine abnehmende oder relativ konstante Klassifikationsgüte zu erkennen. Im arithmetischen Mittel über die Klassen wird bei 25 SVMs die beste Klassifikationsgüte erreicht.

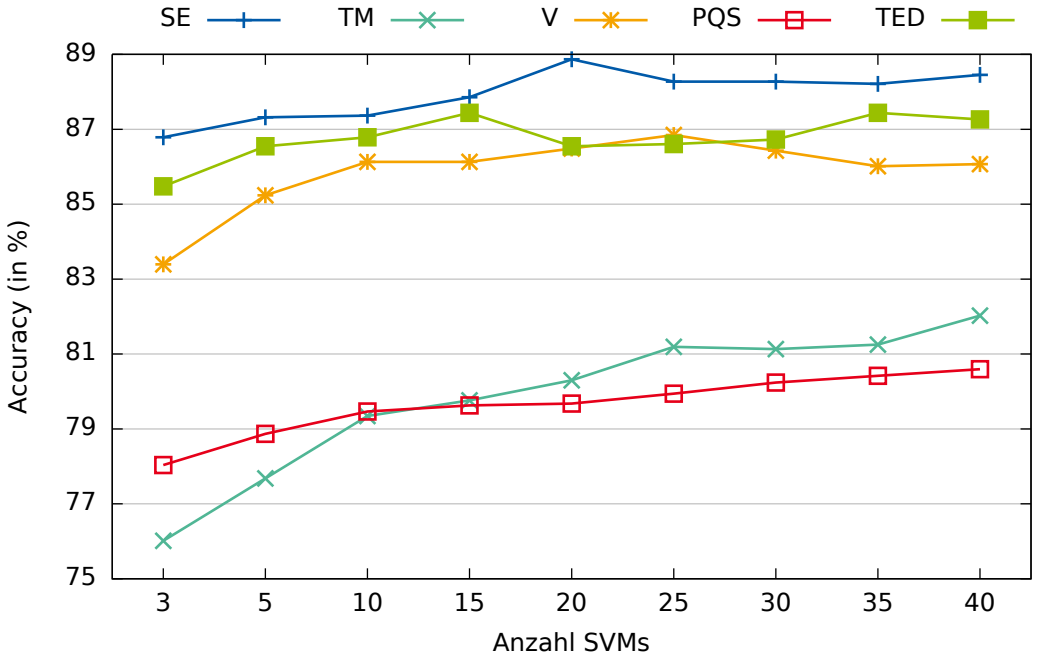


Abbildung 12: Accuracy des CENFA-Verfahrens für die einzelnen Klassen in Abhängigkeit der Zahl der SVMs im Basisklassifikator (Verwendung von 10% der Evaluationsdaten)

Die Klassifikationsgüte bei Verwendung von 25% der Evaluationsdaten ist Abbildung 13 zu entnehmen. Hier ist zunächst ein Anstieg der Klassifikationsgüte zu erkennen, es kommt jedoch zu einer Sättigung der Kurve bei 10-15 SVMs. Für die Klasse *TM* ist ein lokales Maximum bei 20 SVMs zu erkennen.

Abbildung 14 zeigt die Klassifikationsgüte bei Verwendung von 50%, 75% und 100% in Abhängigkeit von der Anzahl der SVMs. Auch hier ist zunächst bei steigender Zahl der SVMs eine Steigerung der Klassifikationsgüte zu erkennen. Bei 50% der Evaluationsdaten ist die Sättigung bei 20 SVMs erreicht, bei 75% bei 15 SVMs und bei 100% bei 10 SVMs. Insgesamt ist aber eine deutliche Abflachung der Kurve bei 10 SVMs zu erkennen.

Auf Basis der in diesem Abschnitt beschriebenen Vorexperimente wurde die Anzahl der SVMs in Abhängigkeit der einzelnen verwendeten Anteile der Evaluationsdaten für die folgenden Experimente festgelegt. Um eine bessere Vergleichbarkeit über die einzelnen Experimente zu gewährleisten, wird dieser Parameter nicht in Abhängigkeit von den jeweiligen Klassen gesetzt. Dies wäre jedoch für eine Nutzung in einem Produktivsystem überlegenswert. Tabelle 20 gibt einen Überblick über die Parameterwahl für die folgenden Experimente. Diese Werte für die Anzahl der SVMs wurden unter Berücksichtigung der in diesem Abschnitt vorgestellten Ergebnisse möglichst klein gewählt, um den Berechnungsaufwand während des Trainings des Basisklassifikators gering zu halten. Während bei 10% der Evaluationsdaten eine deutliche Verbesserung bei Verwendung von 25 SVMs zu erkennen war (im Vergleich

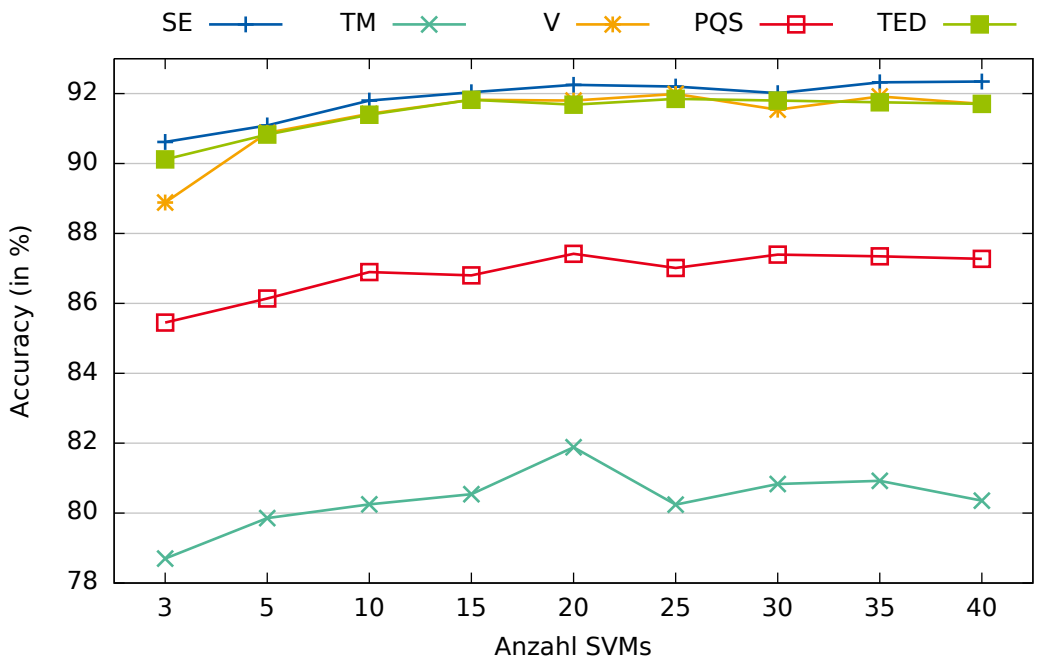


Abbildung 13: Accuracy des CENFA-Verfahrens für die einzelnen Klassen in Abhängigkeit der Zahl der SVMs im Basisklassifikator (Verwendung von 25% der Evaluationsdaten)

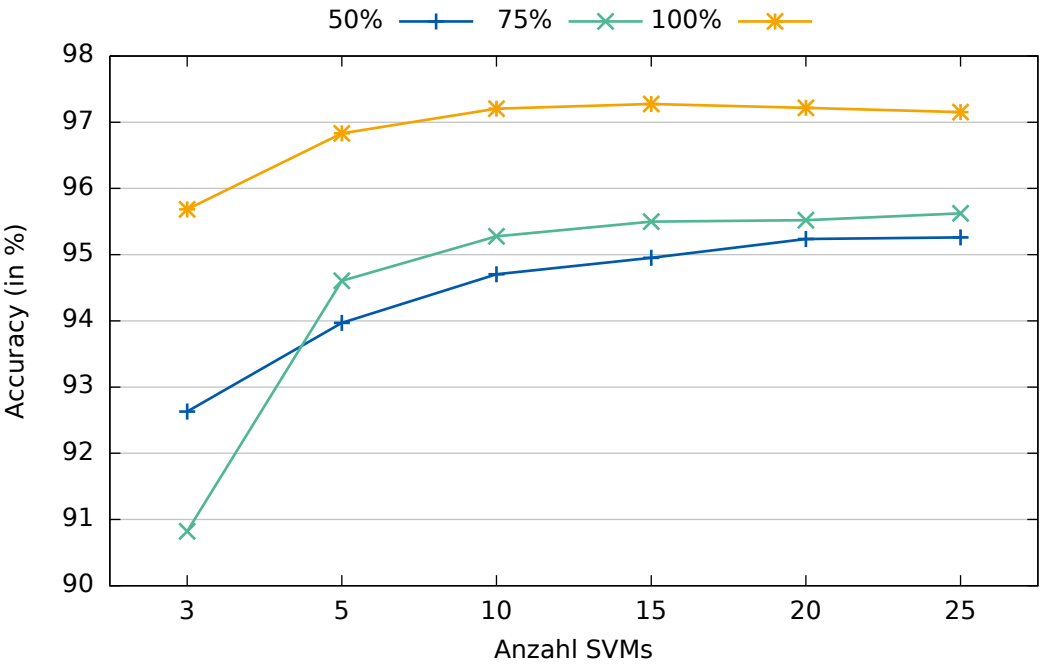


Abbildung 14: Accuracy des CENFA-Verfahrens für die Klasse *SE* in Abhängigkeit der Zahl der SVMs im Basisklassifikator (Verwendung von 50%, 75% und 100% der Evaluationsdaten)

zur Verwendung von weniger SVMs), war für die anderen Größen ab 10 SVMs keine oder nur noch eine geringe durchschnittliche Verbesserung zu erkennen.

Tabelle 20: Übersicht über die gewählte Anzahl an SVMs in Abhängigkeit vom verwendeten Anteil der Evaluationsdaten

Anteil der Evaluationsdaten	Anzahl SVMs
10%	25
25%	10
50%	10
75%	10
100%	10

6.3.5.3 *Überblick über Klassifikationsgüte und Rechenzeit*

Die Klassifikationsgüte für CENFA bei Nutzung unterschiedlicher Prozentsätze der verfügbaren Evaluationsdaten ist Abbildung 15 zu entnehmen. Dargestellt ist die Accuracy für die Klassifikation in die betrachteten Klassen sowie das arithmetische Mittel dieser Werte. Insgesamt lässt sich eine Steigerung der Klassifikationsgüte bei steigender Anzahl an Evaluationsdaten und somit steigender Anzahl an Trainingsdaten beobachten. Mit Ausnahme der Ergebnisse für 25% der Daten für die Klasse *TM* ist eine stetige Steigung erkennbar (vergleiche Erläuterungen zur Lernkurve in Abschnitt 2.2.1). Das arithmetische Mittel der Accuracy steigt von 84,57% auf 96,61%. Diese Steigerung des Mittelwerts ist mit dem insgesamt besseren Training des Systems durch die größere Zahl an Trainingsdokumenten zu erklären.

Weiterhin lässt sich beobachten, dass die Klassifikationsgüte für die einzelnen Klassen unterschiedlich hoch ist. So fällt die Accuracy für die Klassen *SE*, *TED* und *V* durchgehend höher aus als für die Klassen *PQS* und *TM*. Insgesamt gehen die Unterschiede in der Klassifikationsgüte für die einzelnen Klassen bei steigender Korpusgröße zurück. So reduziert sich die Standardabweichung des arithmetischen Mittels von 3,74% bei 10% Korpusgröße auf 1,52% bei 100% Korpusgröße.

Abbildungen 16 beziehungsweise 17 zeigen die Accuracy der vier unterschiedlichen Klassifikationsverfahren für die fünf betrachteten Klassen bei Verwendung von 10% beziehungsweise 100% der Evaluationsdaten und den angegebenen Parametersettings. Eine vollständige Übersicht der Accuracy für alle Korpusgrößen ist im Anhang A.3 in Tabelle 36 gegeben.

Neben der Klassifikationsgüte ist auch die Zeit zum Training eines Klassifikators eine relevante Kenngröße, da diese ausschlaggebend für die Effizienz eines Klassifikators ist. Tabelle 21 zeigt die durchschnittlich benötigte Rechenzeit zum Training der unterschiedlichen Systeme. Angegeben sind jeweils die Rechenzeiten für das erste Training und die folgenden Trainingszyklen. Das erste Training beinhaltet im Falle von CENFA, Random und Extended das Training des Basisklassifikators, also dem Ensemble der Klassifikatoren. Im Falle der RSSVM beinhaltet es das Training der einzelnen SVM. Die weiteren Trainingszyklen beinhalten im Fall von CENFA und Random das Neutraining des Spezialklassifikators, also der einzelnen SVM. Im

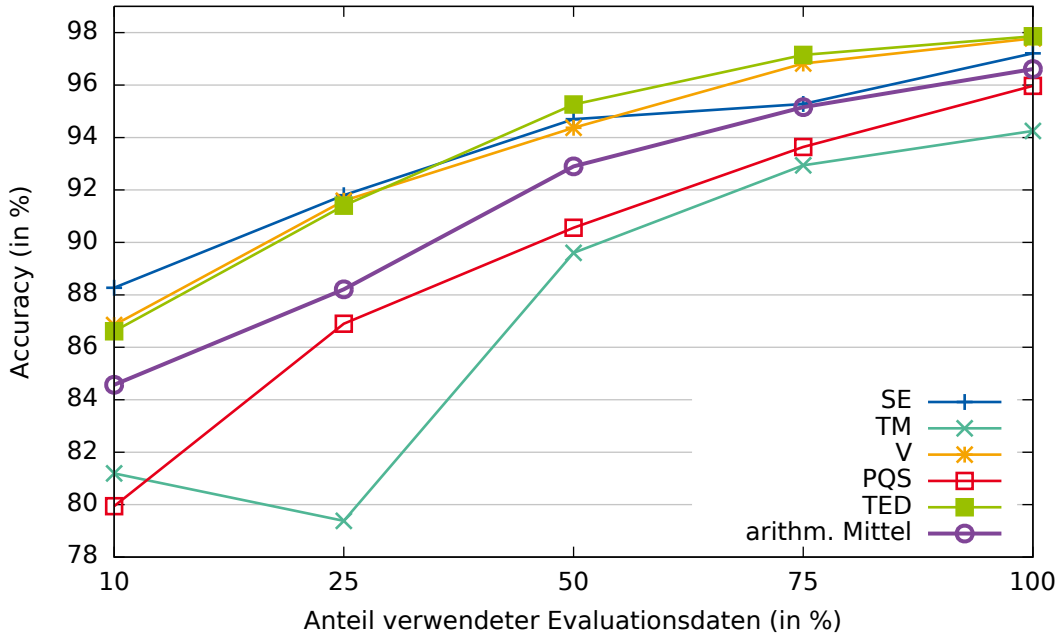


Abbildung 15: Accuracy des CENFA-Verfahrens bei Verwendung unterschiedlicher Prozentsätze der Evaluationsdaten für die einzelnen Klassen

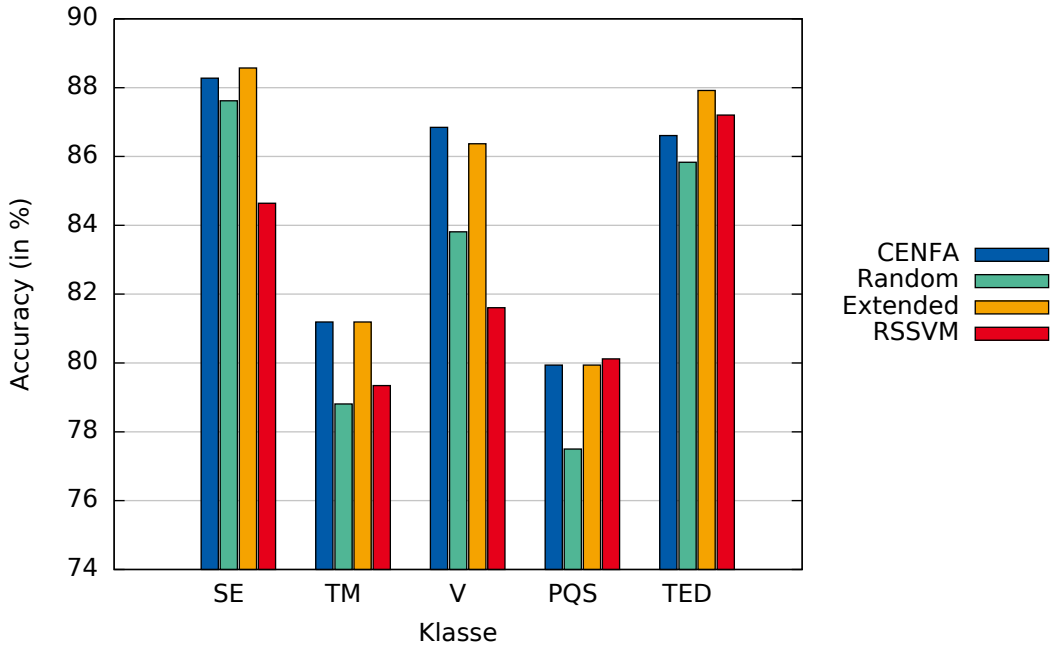


Abbildung 16: Accuracy des CENFA-Verfahrens und der zu vergleichenden Verfahren bei 10% Prozent Korpusgrüße; Anzahl SVMs im Ensemble: 15; AKG = 0,7; AF = 0,7

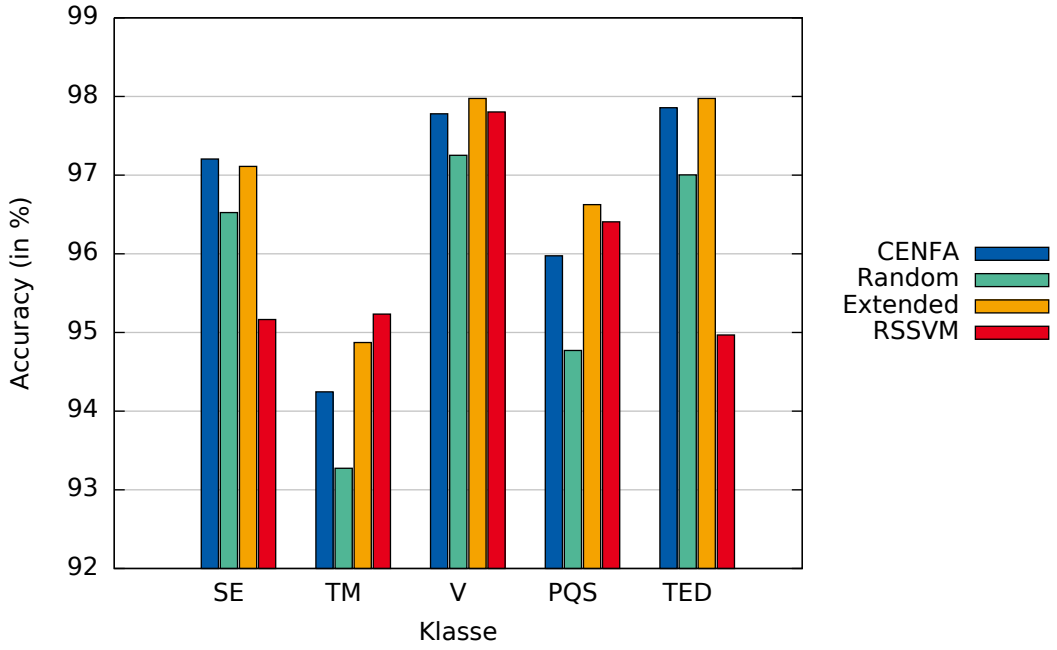


Abbildung 17: Accuracy des CENFA-Verfahrens und der zu vergleichenden Verfahren bei 100% Prozent Korpusgrüße; Anzahl SVMs im Ensemble: 10; AKG = 0,7; AF = 0,7

Falle von Extended beinhalten sie das Neutraining des Ensembles. Bei der RSSVM beinhalten sie das Neutraining der einzelnen SVM, die mit allen bisher verfügbaren Trainingsdokumenten aus den einzelnen Iterationsschritten trainiert wird. Aus Gründen der Übersichtlichkeit sind nur die Korpusgrößen 10%, 50% und 100% dargestellt. Weitere Ergebnisse sind in Anhang A.3, Tabelle 37 zu finden.

Im Folgenden werden die zuvor vorgestellten Ergebnisse der Accuracy und der Rechenzeiten gemeinsam hinsichtlich unterschiedlicher Aspekte diskutiert.

6.3.5.4 *Einfluss der Auswahl ambiguer Instanzen*

Der CENFA-Ansatz zeigt durchgehend eine höhere Accuracy als der Random-Ansatz (siehe Abbildung 16 und Abbildung 17). So liegen bei Verwendung von 10%

Tabelle 21: Übersicht über die benötigten Rechenzeiten in Sekunden für das Training der einzelnen Klassifikatoren, die unterschiedlichen Korpusgrößen und die Phase des Betriebs (init. = das erste Training des Systems, Neutrain. = jedes weitere Training)

	10%		50%		100%	
	INIT.	NEUTRAIN.	INIT.	NEUTRAIN.	INIT.	NEUTRAIN.
CENFA	32,4	<0,1	775,6	<0,1	4.976,1	<0,1
Random	32,4	<0,1	775,6	<0,1	4.976,1	<0,1
RSSVM	1,5	1,5	102,9	102,9	613,3	613,3
Extended	32,4	34,2	775,6	848,5	4.976,4	5.106,6

der Evaluationsdaten die Verbesserungen zwischen 0,7 Prozentpunkten und 3,0 Prozentpunkten. Bei Verwendung aller Evaluationsdaten liegen die Verbesserungen zwischen 0,5 und 1,2 Prozentpunkten. Somit kann bei gleicher Architektur und gleicher Anzahl an Trainingsdaten für das Training der einzelnen Klassifikatoren im CENFA-System eine höhere Klassifikationsgüte erzielt werden. Dies ist auf die Auswahl der ambiguen Instanzen zum Training des Spezialklassifikators zurückzuführen. Abbildung 18 zeigt das arithmetische Mittel der Accuracy-Werte für die beiden Ansätze und alle Evaluationsgrößen. Hier ist zu erkennen, dass für alle Evaluationsgrößen der CENFA-Ansatz dem Random-Ansatz hinsichtlich der Klassifikationsgüte überlegen ist. Bei Betrachtung der beispielhaft eingezeichneten gestrichelten Linie lässt sich erkennen, dass der CENFA-Ansatz zum Erreichen einer spezifischen Accuracy weniger Trainingsdaten benötigt als der Random-Ansatz. Die Rechenzeiten für das Training der beiden Ansätze sind identisch, da sich die Architektur und die Anzahl der verwendeten Trainingsdaten für die beiden Ansätze nicht unterscheiden.

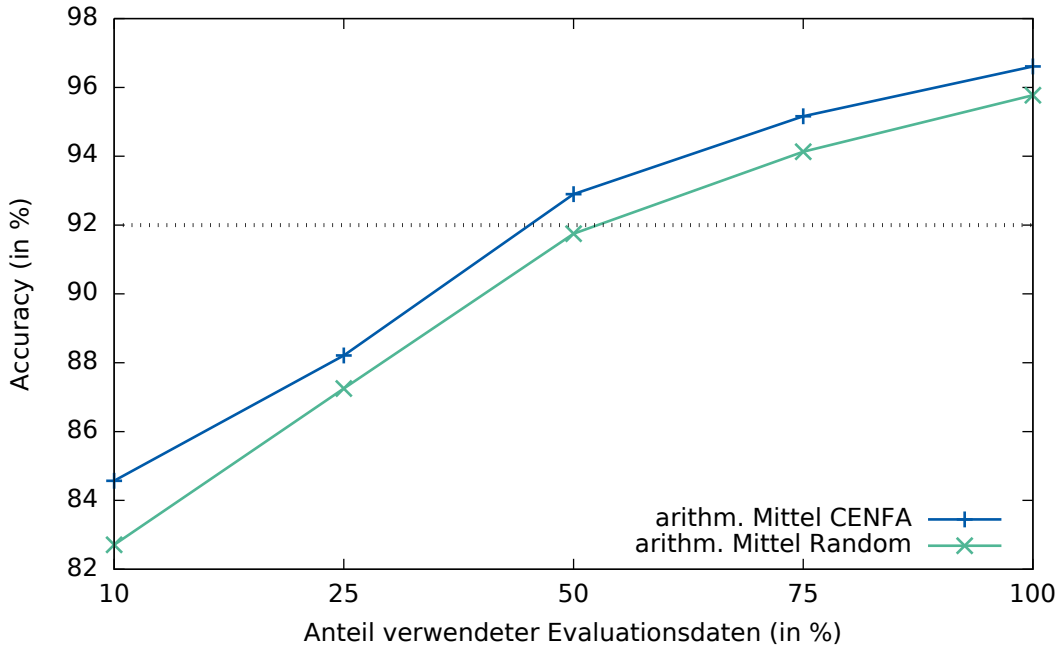


Abbildung 18: Arithmetisches Mittel der Accuracy für CENFA und *Random* bei Verwendung verschiedener Anteile des Evaluationskorpus

6.3.5.5 *Einfluss des Neutrainings des Spezialklassifikators*

Beim Vergleich der Klassifikationsgüte von CENFA und Extended bei 10% Korpusgröße (Abbildung 16) zeigen sich nur kleine Unterschiede. So ist bei 10% Korpusgröße die Klassifikationsgüte von CENFA und Extended in den beiden Klassen *PQS* und *TM* identisch, bei den Klassen *TED* und *SE* weist die Accuracy für den Extended-Ansatz 1,3 beziehungsweise 0,3 Prozentpunkte höhere Werte auf, bei der Klasse *V* ist die Accuracy für CENFA 0,5 Prozentpunkte größer. Das arithmetische Mittel der einzelnen Accuracy-Werte liegt bei CENFA bei 84,6% und bei Extended bei 84,8%. Bei 100% Korpusgröße zeigt der Extended-Ansatz bei den vier Klassen *PQS*, *TED*, *TM*, *V* um maximal 0,6 Prozentpunkte bessere Werte. Bei der Klasse *SE* ist CENFA

minimal überlegen (um <0,1 Prozentpunkte). Im arithmetischen Mittel ist die Accuracy für 100% Korpusgröße für Extended um 0,2 Prozentpunkte besser als für CENFA. Zusammenfassend lässt sich sagen, dass die Klassifikationsgüte somit bei dem Vergleichsansatz Extended bei 10% und bei 100% Korpusgröße geringfügig besser ausfällt als beim CENFA-Verfahren.

Bei Analyse von Tabelle 21 für den Vergleich zwischen CENFA und Extended ist folgendes zu erkennen: Das Neutraining des Systems erfolgt beim Extended-Ansatz deutlich langsamer als beim CENFA-Ansatz. Während bei 10% Korpusgröße ein um den Faktor 2.441 langsameres Neutraining erfolgt, ist für das Neutraining bei 100% Korpusgröße ein 100.130-fach größerer Zeitbedarf notwendig. Das initiale Training benötigt bei beiden Verfahren gleich lang, da in beiden Fällen das komplette Ensemble trainiert werden muss. Um die Entwicklung dieses relativen Zeitbedarfs zu sehen, wird im Folgenden der Zeitfaktor definiert:

$$\text{Zeitfaktor}(x) = \frac{\text{Zeitbedarf}(x)}{\text{Zeitbedarf}(\text{CENFA})}. \tag{8}$$

Abbildung 19 zeigt diesen Zeitvorteil für die unterschiedlichen Prozentsätze der verwendeten Evaluationsdaten (x = Extended). Insgesamt ist also CENFA dem *Extended*-Ansatz hinsichtlich des Zeitbedarfs deutlich überlegen.

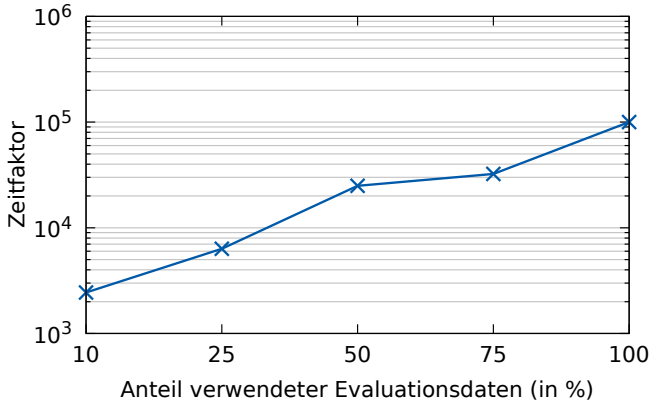


Abbildung 19: Zeitaufwand für das Neutraining des Extended-Ansatzes im Vergleich zum Zeitaufwand für das Neutraining des CENFA-Ansatzes bei Verwendung unterschiedlicher Anteile der Evaluationsdaten

6.3.5.6 *Einfluss der Nutzung des Ensemble Learning*

Beim Vergleich der Klassifikationsgüte von CENFA und RSSVM (siehe Abbildung 16 und Abbildung 17) lässt sich keine eindeutige Aussage treffen. Bei 10% Korpusgröße ist CENFA in Hinsicht auf die drei Klassen *SE*, *TM* und *V* überlegen und erreicht hier um 1,9 (*TM*) bis 5,2 Prozentpunkte (*V*) bessere Ergebnisse. Bei den verbleibenden beiden Klassen ist jedoch RSSVM um 0,2 (*PQS*) beziehungsweise 0,6 Prozentpunkte (*TED*) geringfügig überlegen. Bei 100% Korpusgröße ist CENFA bei zwei Klassen überlegen um 2,1 (*SE*) beziehungsweise 2,9 Prozentpunkte (*TED*). RSSVM zeigt eine höhere Klassifikationsgüte bei den verbleibenden drei Klassen: um 0,4 Prozentpunkte bei *PQS*, 1,0 Prozentpunkte bei *TM* und um 0,1 Prozentpunkte bei *V*. Im

artithmetischen Mittel ist CENFA sowohl bei Verwendung von 10% der Evaluationsdaten (84,6% gegenüber 82,6% Accuracy) als auch bei Verwendung von 100% der Evaluationsdaten (96,6% gegenüber 95,9%) überlegen.

Beim Vergleich der Rechenzeiten für das Training von CENFA und RSSVM (Tabelle 21) fällt auf, dass das initiale Training für den einzelnen Klassifikator im RSSVM deutlich schneller vonstatten geht als für das Ensemble im Basisklassifikator des CENFA-Ansatzes (Faktor 8 für 100% Korpusgröße bis Faktor 22 für 10% Korpusgröße). Beim erneuten Training ist jedoch CENFA deutlich überlegen mit Faktor 107 bei 10% Korpusgröße bis hin zu Faktor 12.026 bei 100% Korpusgröße. Der steigende Zeitvorteil von CENFA bei steigenden Korpusgrößen für das Neutraining unter Verwendung von Formel 8 mit $x = \text{RSSVM}$ ist in Abbildung 20 zu sehen.

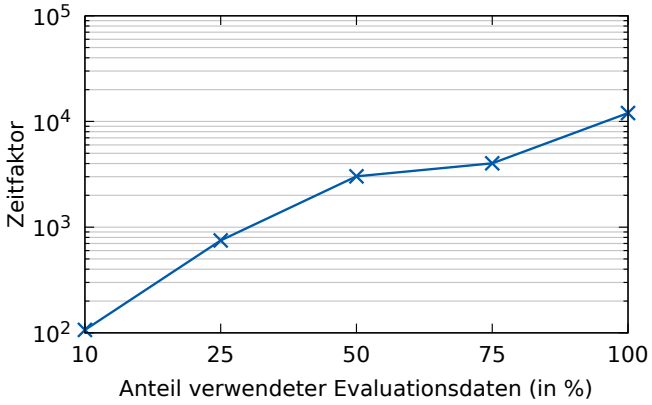


Abbildung 20: Zeitaufwand für das Neutraining des RSSVM-Ansatzes im Vergleich zum Zeitaufwand für das Neutraining des CENFA-Ansatzes bei Verwendung unterschiedlicher Anteile der Evaluationsdaten

6.4 FAZIT

Das vorgestellte Verfahren CENFA (Abschnitt 6.2) kombiniert Ansätze des Active Learning und des Ensemble Learning mit dem Ziel eines robusten Klassifikators für Texte bei einem reduzierten Bedarf an annotierten Trainingsdokumenten. Das vorgestellte Verfahren besteht aus zwei Klassifikationskomponenten: dem Basisklassifikator, der aus einem Ensemble von SVMs besteht und nur initial trainiert wird, sowie dem Spezialklassifikator, der aus einer einzelnen SVM besteht und bei Bedarf neu trainiert werden kann. Zum Neutraining des Spezialklassifikators werden vom Basisklassifikator aus den zu klassifizierenden Instanzen die als ambigue erscheinenden Instanzen bestimmt, welche von einem menschlichen Annotator annotiert werden.

Das Verfahren wurde unter Nutzung eines Evaluationskorpus aus der Domäne *Stellenanzeigen* und dem Vergleich zu anderen Ansätzen evaluiert. Der Vergleich zum Random-Ansatz zeigt den Vorteil der Verwendung ambiguer Instanzen zum Training des Spezialklassifikators hinsichtlich der Klassifikationsgüte (bis zu 3 Prozentpunkte Verbesserung der Accuracy) und die daraus resultierende gute Klassifikationsgüte bei Verwendung einer reduzierten Anzahl an annotierten Trainingsdaten. Der Vergleich zum RSSVM-Ansatz zeigt einen leichten Vorteil der Verwendung eines Ensembles im Vergleich zur Nutzung eines einzelnen Klassifikators, was im Kontrast zu den Untersuchungen von Dong und Han [41] steht, die bei einer Text-

klassifikationsaufgabe keinen Mehrwert durch die Verwendung von Bagging erkennen. Insbesondere zeigt sich aber der geringere Zeitbedarf für ein Neutrainieren des Systems im Vergleich zur Verwendung einer einzelnen SVM (Verbesserung um maximal Faktor 12.000). Dieser Vorteil hinsichtlich des Zeitbedarfs wird auch im Vergleich mit dem Extended-Ansatz deutlich, welcher zeigt, dass aus dem Neutrainieren eines Ensembles eine erhebliche Zeitanforderung resultiert (Verbesserung durch Nutzung von CENFA um bis zu Faktor 100.000). Die Trennung in Basisklassifikator und Spezialklassifikator führt jedoch zu einer geringfügigen schlechteren Klassifikationsgüte (maximal 1,3 Prozentpunkte schlechter).

Insgesamt konnte also gezeigt werden, dass sich CENFA durch eine gute Kombination zwischen einer hohen Klassifikationsgüte, einem guten Zeitbedarf und einem reduzierten Bedarf an manuell zu annotierenden Trainingsdaten auszeichnet. Das Erreichen einer guten Klassifikationsgüte mit weniger Trainingsdokumenten als klassische Verfahren macht das Verfahren geeignet zur Identifikation von Meta-Attributen.

7

IDENTIFIKATION VON AGGREGIERTEN ATTRIBUTEN

Aggregierte Attribute zeichnen sich durch ihre sequentielle Struktur, bestehend aus mehreren atomaren Attributen (vergleiche Abschnitt 4.3), aus. Die Identifikation eines aggregierten Attributes ist nur von Wert, wenn alle atomaren Attribute, also die einzelnen Elemente der Sequenz, erfolgreich identifiziert werden können. Dies stellt besonders hohe Anforderungen an das Verfahren zur Identifikation. Andererseits kann die sequenzielle Struktur jedoch auch genutzt werden, um die atomaren Attribute unter Verwendung ihrer externen Struktur, also den anderen Elementen der Sequenz, eindeutig zu identifizieren. Dieser Vorteil wird in dem im Folgenden vorgestellten Verfahren genutzt. Das Verfahren wird zunächst abstrakt dargestellt (Abschnitt 7.1) und dann anhand des Anwendungsfalls der Identifikation postalischer Unternehmensadressen konkretisiert (Abschnitt 7.2) und evaluiert (Abschnitt 7.3) [147].

7.1 KONZEPT

Das vorgestellte Verfahren besteht aus drei wesentlichen Schritten, welche im Folgenden skizziert werden.

In einem ersten Schritt wird das komplette zu strukturierende Dokument vorverarbeitet, um die anschließende Identifikation atomarer Attribute zu ermöglichen. Die Vorverarbeitung besteht aus mehreren Teilschritten. Dies sind eine optionale Säuberung der Daten, eine Tokenisierung des Texts und weiterhin eine automatische Annotation der einzelnen Token mit relevanten Merkmalen für die folgenden Identifikationsschritte.

Im zweiten Schritt erfolgt die Identifikation von Kandidaten für die atomaren Attribute. *Attributskandidaten* sind Tokensequenzen, welche die Anforderungen an ein Attribut erfüllen, aber nicht zwangsläufig Teil des finalen aggregierten Attributs sind. Ein Attributskandidat ist kein Teil des aggregierten Attributs, falls sich keine weiteren atomaren Attribute identifizieren lassen, die diesem atomaren Attribut zugeordnet werden können, um ein aggregiertes Attribut zu bilden. Zur Identifikation der atomaren Attribute können Abhängigkeiten zwischen den einzelnen Attributskandidaten genutzt werden, so kann beispielsweise in der textuellen Umgebung eines Attributskandidaten nach der Existenz eines anderen atomaren Attributs gesucht werden. Aus solchen Abhängigkeiten ergibt sich eine Ausführungsreihenfolge der einzelnen Identifikationsschritte. Es sollten dabei zunächst die Attribute identifiziert werden, von deren Identifikation ein hoher Recall erwartet wird, um mögliche Folgefehler durch nicht identifizierte Attribute zu reduzieren.

Im dritten und letzten Schritt werden die einzelnen Attributskandidaten zu einem aggregierten Attribut zusammengefasst. Hierbei wird von der Menge an Attributskandidaten eines spezifischen Attributs ausgegangen und für jedes der Elemente versucht, weitere passende atomare Attribute auf Basis ihrer Kandidaten auszuwäh-

len. Sobald ein Attributskandidat als Element eines aggregierten Attributs erkannt wurde, kann dieses nicht mehr zur Bildung eines weiteren aggregierten Attributs verwendet werden.

7.2 ANWENDUNGSFALL: IDENTIFIKATION POSTALISCHER UNTERNEHMENSADRESSEN

Im vorigen Abschnitt wurde ein generisches Konzept zur Identifikation aggregierter Attribute skizziert. Dieses muss für die Anwendung in einer Domäne für die jeweils zu identifizierenden aggregierten Attribute konkretisiert werden. Insbesondere müssen die enthaltenen atomaren Attribute definiert, das Vorgehen der Identifikation von Kandidaten für diese atomaren Attribute festgelegt werden und definiert werden, wie die einzelnen Attributskandidaten aggregiert werden.

In diesem Abschnitt wird eine solche Anpassung am Beispiel der Identifikation postalischer Adressen von Unternehmen vorgestellt. Der Fokus wird auf Adressen deutscher Unternehmen gelegt. Auch wenn standardisierte Regeln für die Schreibweise deutscher Adressdaten in Form einer DIN-Norm existieren [118], sind Adressdaten häufig nicht gemäß dieser Regeln repräsentiert. Häufige Abweichungen, gerade in Webdokumenten, sind das Fehlen von Zeilenumbrüchen, die inkorrekte Kapitalisierung von Eigennamen oder nicht zur Adresse gehörige Wortsequenzen zwischen den einzelnen atomaren Attributen einer Adresse. Weiterhin kann es bei Konvertierung der ursprünglichen Dokumentenrepräsentation, wie einer HTML-Seite, zu einem Rohtextdokument zu Defekten in der Ordnung der einzelnen Wörter kommen, beispielsweise bei tabellarischen Repräsentationen. Diese Tatsachen erfordern eine hohe Robustheit des Verfahrens zur Identifikation von Adressen, um trotzdem eine hohe Güte erzielen zu können. Aus diesem Grund können keine strikten Regeln zur Identifikation verwendet werden, welche Abweichungen in der Schreibweise nicht tolerieren.

Die Identifikation von Unternehmensadressen ist von hoher Relevanz, da mit den resultierenden Daten beispielsweise automatisiert Branchenverzeichnisse erstellt werden können oder die automatisierte Population geographischer Informationssysteme mit Unternehmensdaten ermöglicht wird. Im Rahmen dieser Arbeit werden Unternehmensadressen in der Domäne der *Impressumsseiten* adressiert (siehe Abschnitt 4.2.2). Tabelle 22 zeigt die atomaren Attribute, welche im Rahmen dieser Arbeit als Elemente deutscher Unternehmensadressen angenommen werden.

Das Format von Adressen unterscheidet sich von Land zu Land, dies betrifft sowohl die Struktur des aggregierten Attributs als auch die Struktur der atomaren Attribute. Zur Anwendung des Ansatzes in einem anderen Land müssen somit sowohl die gesamte Struktur angepasst werden als auch die Regeln zur Identifikation der atomaren Attribute.

Eine Übersicht über die einzelnen Schritte zur Identifikation postalischer Adressen ist in Abbildung 21 dargestellt. Im Folgenden wird auf die einzelnen Schritte eingegangen. Die maximalen Tokenabstände, die während der Identifikation und Aggregation der Attributskandidaten bei Abhängigkeiten zwischen einzelnen Attributskandidaten angenommen werden, sind als feste Parameter gesetzt und werden nicht variiert. Diese Parameter wurden auf Basis einer manuellen Analyse einiger Beispieldokumente bestimmt.

Tabelle 22: Atomare Attribute innerhalb des aggregierten Attributs *Unternehmensadresse*

ATTRIBUT	ATTRIBUTWERTE (BEISPIELE)	ATTRIBUTTYP
Unternehmensname	„Robert Bosch GmbH“, „Schmidt & Partner“	Eigenname
Straßenname	„Rheinstr.“, „Große Bleiche“, „Wacholderweg“	Eigenname
Hausnummer	„45“, „39-45“	numerisches Attribut
Postleitzahl	„64283“, „35398“	numerisches Attribut
Städtename	„Frankfurt a.M.“, „Darmstadt“, „Linden"	Eigenname

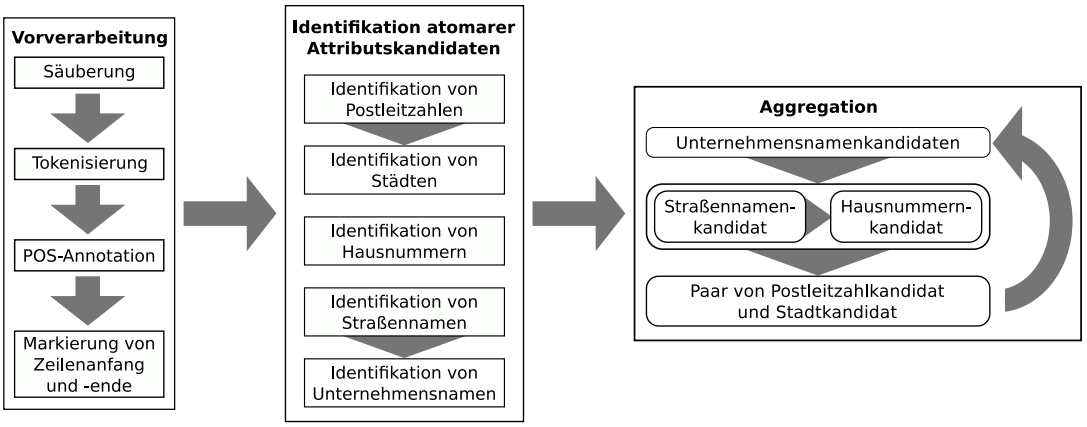


Abbildung 21: Übersicht über den Ansatz zur Identifikation von aggregierten Attributen am Beispiel der Identifikation postalischer Unternehmensadressen; Rechtecke repräsentieren Prozessschritte, Rechtecke mit abgerundeten Ecken repräsentieren Attributskandidaten, graue Pfeile repräsentieren die Reihenfolge der Prozessschritte und graue Dreiecke repräsentieren Abhängigkeiten zwischen Attributskandidaten und die sich daraus ergebende Reihenfolge bei der Identifikation

7.2.1 *Vorverarbeitung*

Während der Vorverarbeitung werden die Dokumente in ein Format überführt, welches zur anschließenden Identifikation der atomaren Attributskandidaten verwendet wird. Dafür erfolgt zunächst eine Säuberung des Texts, bei der Zeichen entfernt werden, welche nicht Unicode-konform [168] sind. Weiterhin werden Leerzeichen zwischen Ziffern entfernt. Dies soll verhindern, dass bei der folgenden Tokenisierung beispielsweise Postfachnummern, welche zur besseren Lesbarkeit Leerzeichen enthalten können, auf mehrere Token aufgeteilt werden. Die aus der Tokenisierung resultierenden Token werden jeweils mit einem Merkmalsvektor angereichert. Die Elemente in diesem Vektor sind in Tabelle 23 dargestellt. Abgesehen vom Merkmal `POS-Tag` sind alle Merkmale boolesche Werte. Das Merkmal `Zeilenanfang` wird auf `True` gesetzt, wenn dem Token ein Zeilenumbruch vorausgeht, ansonsten wird es mit `False` belegt. Analog wird das Merkmal `Zeilenende` gesetzt, wenn dem Token ein Zeilenumbruch folgt. Das Merkmal `POS-Tag` wird mit der Wortart belegt, die

durch einen POS-Tagger bestimmt wird. Alle weiteren Merkmale werden zunächst mit `False` initialisiert und können während der Identifikation der atomaren Attribute alternativ gesetzt werden. Verwendet werden sie während der Identifikation der atomaren Attribute und auch nachfolgend während der Aggregation der Attributskandidaten.

7.2.2 *Identifikation atomarer Attributskandidaten*

Die Identifikation atomarer Attributskandidaten ist aufgeteilt in die Identifikation der Kandidaten für die einzelnen Attribute. Dabei hängt die Identifikation der einzelnen Attribute teilweise voneinander ab, wodurch sich die Reihenfolge für die Ausführung ergibt. Die Abhängigkeiten wurden so gewählt, dass als Ausgangspunkt jeweils das Attribut verwendet wird, bei dessen Identifikation der höchste Recall (siehe Formel 4, Abschnitt 2.2.3.2) erwartet wird. Durch den Fokus auf den Recall soll erreicht werden, dass möglichst wenige Kandidaten für die initialen Attribute nicht erkannt werden. Eine hohe Fehlerrate würde sich bei der Identifikation der abhängigen Attribute weiter fortpflanzen. Als Ausgangspunkt wird zunächst das Attribut *Postleitzahl* genutzt, da aufgrund der geringen Heterogenität der internen Struktur von Postleitzahlen ein hoher Recall erwartet wird. In der textuellen Umgebung der Postleitzahlenkandidaten wird dann nach Städtenamen gesucht. Weiterhin werden, ausgehend von Kandidaten für die Attribute *Hausnummer* und *Straßenname*, Kandidaten für das Attribut *Unternehmensname* gesucht.

Zur Identifikation der atomaren Attribute *Straßenname* und *Stadt* wird im vorgestellten Verfahren auf die Wissensbasis OpenStreetMap (OSM) zurückgegriffen. OSM[1] [62] ist ein kollaboratives Projekt mit dem Ziel der Erstellung frei verfügbaren Kartenmaterials. Durch die Veröffentlichung unter der Open Database License (OBdL) 1.0 wird jegliche Nutzung der Daten ermöglicht, was einen Mehrwert im Gegensatz zu proprietären Systemen wie Google Maps[2] oder TomTom[3] bietet. Es wurde gezeigt, dass die Abdeckung und Qualität der OSM-Daten in Deutschland ähnlich gut ist wie bei proprietären Systemen [115]. Die Abdeckung wird in anderen Ländern ähnlich hoch eingeschätzt [59]. Durch unterschiedliche Schnittstellen lassen sich spezifische geographische Elemente, wie Straßennamen oder Städtenamen, durch Verwendung von Abfragesprachen gezielt auswählen, was eine effiziente Nutzung als Datenquelle ermöglicht. Diese Faktoren würden die Verwendung von OSM auch zur Identifikation von Adressen in anderen Ländern ermöglichen.

7.2.2.1 *Identifikation von Postleitzahlen*

Deutsche Postleitzahlen folgen einem strengen Muster. Sie bestehen aus fünf Ziffern, welchen optional „D-“ vorgestellt wird, um von Postleitzahlen anderer Länder abzugrenzen. Diese homogene interne Struktur des Attributs ermöglicht das Verwenden eines regulären Ausdrucks zur Identifikation. Es wird folgender regulärer Ausdruck verwendet, der das beschriebene Muster abbildet: `(D-)?[0-9]{5}`. Für alle Token, die durch diesen regulären Ausdruck beschrieben werden können, wird das Merkmal `Postleitzahl` auf `True` gesetzt.

1 https://www.openstreetmap.org, letzter Zugriff am 18.09.2015
2 http://maps.google.com, letzter Zugriff am 18.09.2015
3 http://www.tomtom.com, letzter Zugriff am 18.09.2015

Tabelle 23: Übersicht über die verwendeten Merkmale zur Annotation von Token bei der Identifikation von postalischen Adressen

Merkmal	Bedeutung
`POS-Tag`	Wortart des Token
`Zeilenanfang`	Token ist der Anfang einer Zeile
`Zeilenende`	Token ist das Ende einer Zeile
`StadtPerListe`	Stadtnamenkandidat wurde auf Basis einer Liste identifiziert
`StadtPerStruktur`	Stadtnamenkandidat wurde auf Basis der externen Struktur identifiziert
`Postleitzahl`	Token ist ein Postleitzahlenkandidat
`Hausnummer`	Token ist ein Hausnummernkandidat
`Postfachnummer`	Hausnummer stellt ein Postfachnummernkandidat dar
`StraßeAnfang`	Token ist der Anfang eines Straßennamenkandidaten
`StraßeEnde`	Token ist das Ende eines Straßennamenkandidaten
`StraßeSuffix`	Straßennamenkandidat wurde auf Basis eines Straßennamensuffixes identifiziert
`StraßePOSMuster`	Straßennamenkandidat wurde auf Basis des POS-Musters identifiziert
`StraßePostfach`	Straßennamenkandidat ist ein Postfach
`UnternehmensAnfang`	Token ist der Anfang eines Unternehmensnamenkandidaten
`UnternehmensEnde`	Token ist das Ende eines Unternehmensnamenkandidaten
`UnternehmenPerIndikator`	Unternehmsnamenkandidat wurde per Rechtsform identifiziert
`UnternehmenPerStruktur`	Unternehmensnamenkandidat wurde auf Basis der externen Struktur identifiziert

7.2.2.2 *Identifikation von Städten*

Städtenamen in Deutschland zeichnen sich durch eine hohe Heterogenität in der internen Struktur aus. Sie können sowohl aus einem einzelnen Wort als auch einer Wortsequenz bestehen. Weiterhin werden Städenamen häufig abgekürzt, was eine Vielzahl an Ausdrucksmöglichkeiten für eine spezifische Stadt erlaubt. Die Wortsequenzen „Frankfurt am Main", „Frankfurt", „Ffm", „Frankfurt a.M." sind beispielsweise alles Synome für die gleiche Stadt. Auch eine Ambiguität ist bei Städenamen gegeben. Zum einen sind Städtenamen häufig nicht eindeutig, beispielsweise gibt es mehrere Städte mit dem Namen „Neustadt" oder „Frankfurt", zum anderen gibt es Städtenamen mit alternativen Bedeutungen, so sind „Essen" oder „Gießen" sowohl Städtenamen als auch substantivierte Verben.

Aufgrund der beschriebenen Heterogenität des Attributs *Städtename* können keine unflexiblen Regeln, wie sie durch reguläre Ausdrücke definiert werden, verwendet werden. Die Identifikation von Städtenamen erfolgt durch zwei unterschiedliche, sich ergänzenden Ansätze:

1. Eine Tokensequenz, welche identisch zum Element in einer Liste von Städtenamen ist und in der textuellen Umgebung eines Postleitzahlenkandidaten liegt, wird als Kandidat für einen Städtenamen angenommen.

2. Tokensequenzen, die direkt auf einen Postleitzahlenkandidaten folgen, werden ebenso als Kandidaten für Städtenamen verwendet.

Während der erste Ansatz auf Städtenamen mit regulärer Struktur abzielt, zielt der zweite Ansatz auf das Erkennen variierender Schreibweisen ab. Für den ersten Ansatz wird zunächst unter Zugriff auf OSM, eine Liste aller Städtenamen in Deutschland erstellt. Dies kann vor der eigentlichen Identifikation geschehen, da diese Liste nicht abhängig von den betrachteten Dokumenten ist. Für jeden Postleitzahlenkandidaten wird in der textuellen Umgebung innerhalb von drei Token um den Postleitzahlenkandidaten nach Übereinstimmung mit Elementen dieser Liste von Städtenamen gesucht. Falls eine Übereinstimmung gefunden wird, wird das Merkmal `StadtPerListe` auf `True` gesetzt. Falls für einen Postleitzahlenkandidaten auf diesem Weg kein Kandidat für den Städtenamen identifiziert werden kann, kommt der zweite Ansatz zur Verwendung. Dabei werden Token, die direkt auf den Postleitzahlenkandidaten folgen, betrachtet. Wenn diese nur aus Buchstaben bestehen und der erste Buchstabe großgeschrieben ist, werden sie als zugehöriger Städtenamen angenommen (`StadtPerStruktur=True`). Die Suche in einer größeren textuellen Umgebung würde die Fehlerquote erhöhen, da ein größerer Anteil inkorrekter Städtenamenkandidaten identifiziert werden würde.

7.2.2.3 *Identifikation von Hausnummern*

Hausnummern bestehen in der Regel aus einer Zahl. Wenn eine Adresse mehrere Hausnummern umfasst, tauchen jedoch auch Bereiche von Hausnummern auf. Solche Bereiche sind durch einen Bindestrich dargestellt[4]. Weiterhin können an die Zahlen Suffixe angehängt sein, um mehrere Parteien in einem Haus zu separieren oder falls sich mehrere Gebäude auf einem einzelnen Grundstück befinden. Diese

4 Ein Beispiel für einen Hausnummernbereich ist „45-47".

Suffixe bestehen meist aus einem einzelnen Buchstaben, können aber auch aus einem Buchstaben und einer weiteren Ziffer bestehen[5].

Aufgrund dieser regelmäßigen internen Strukturen wird, analog zur Identifikation von Kandidaten für das numerische Attribut *Postleitzahl*, ein regulärer Ausdruck verwendet. Es kommt folgender regulärer Ausdruck zum Einsatz, der sowohl Hausnummernbereiche als auch Hausnummern mit Suffixen zulässt: `([0-9]{1,3})([a-zA-Z][0-9]?)?(([+|-])([0-9]{1,3})([a-zA-Z][0-9]?)?)?`. Bei Token, die diesem regulären Ausdruck entsprechen, wird das Merkmal `Hausnummer=True` gesetzt.

Als Spezialfall des Attributs *Hausnummer* werden Postfachnummern behandelt. Postfachnummern bestehen aus vier bis acht Ziffern. Zur Identifikation der Kandidaten wird der reguläre Ausdruck *([0-9]{4,8})* verwendet. Falls ein Kandidat über diese Sonderregel per Postfachnummer identifiziert wurde, wird er für die spätere Aggregation entsprechend markiert (`Postfachnummer=True`).

7.2.2.4 *Identifikation von Straßennamen*

Ähnlich zum Attribut der *Städtenamen* zeichnet sich das Attribut *Straßenname* durch eine sehr hohe Heterogenität der Attributwerte aus. Straßennamen können aus einem einzelnen Wort oder aus einer Wortsequenz bestehen. Die Wortsequenzen sind häufig Phrasen, welche durch eine Präposition eingeleitet werden[6]. Auffällig ist das häufige Auftauchen einer Menge von Suffixen (Wortendungen)[7]. Ebenso analog zu Städtenamen ist die Häufigkeit von Variationen der Schreibweise insbesondere hinsichtlich der Verwendung von Abkürzungen; es werden sowohl Suffixe[8] abgekürzt als auch Infixe (Wortmittelteile)[9].

Die Variationen der Schreibweise verbunden mit der großen Zahl an Straßennamen in einem Land machen einen listenbasierten Ansatz, bei dem einzelne Token und Tokensequenzen auf ihr Vorkommen in Listen von Straßennamen überprüft werden, unpraktikabel. Aus diesem Grund werden zwei heuristische Ansätze zur Identifikation von Kandidaten für das Attribut *Straßenname* verwendet.

Der erste Ansatz zur Kandidatenidentifikation baut auf der Beobachtung auf, dass gewisse Straßennamensuffixe vergleichsweise häufig vorkommen. Für diesen Ansatz wird zunächst auf Basis der OSM-Datenbasis eine Liste aller Straßennamen in Deutschland erstellt. Nach der Entfernung von Duplikaten enthält diese Liste 306.575 Straßennamen[10]. Unter Verwendung dieser Liste werden dann die 25 häufigsten Suffixe von deutschen Straßennamen, die aus drei bis zehn Buchstaben bestehen, identifiziert. Die resultierende Liste dieser Suffixe befindet sich in Anhang A.4 in Tabelle 38. Insgesamt enden 213.642 (69,7%) der Straßennamen auf eine dieser 25 häufigsten Suffixe. Zu der Liste wurde noch der Suffix „str." hinzugefügt, da dieser häufig am Ende eines Straßennamen als Abkürzung auftaucht. Wenn ein Token im Dokument auf einen der Suffixe in der Liste endet, wird das Token als Ende eines Straßennamenkandidaten angenommen (`StraßeEnde = True`). Ausgehend von diesem Ende

5 Beispiele für Hausnummern mit Suffixen sind „45a" oder „45a1".
6 Beispiele für Phrasen als Straßennamen sind „An der alten Eiche" oder „Neben dem Mühlweg".
7 Beispiele solcher Suffixe sind „-straße", „-allee" oder „-gasse".
8 Ein häufiges Beispiel für einen abgekürzten Suffix ist „-str." statt „-straße".
9 Beispiele für die Abkürzung von Infixen sind „Bgm.-Jung-Weg" statt „Bürgermeister-Jung-Weg" oder „K.-Adenauer-Str." statt „Konrad-Adenauer-Straße".
10 Die Statistiken beziehen sich auf den OSM-Datensatz von März 2013.

des Straßennamenkandidaten wird der Anfang des Kandidaten gesucht. Dazu werden die vorgehenden Token untersucht. Der Anfang des Straßennamenkandidaten wird am nächsten Token angenommen, welcher am Zeilenanfang steht, aus einem Sonderzeichen oder einer Zahl besteht, Teil eines anderen Attributskandidaten ist oder mehr als drei Token vom Ende des Straßennamenkandidaten entfernt ist. Das Merkmal `StraßenAnfang` wird für dieses Token dann auf `True` gesetzt. Die Tokensequenz zwischen Anfang und Ende wird als Straßennamenkandidat angenommen und für alle enthaltenen Token wird `StraßeSuffix` auf `True` gesetzt.

Für die zweite Heuristik wird das Merkmal `POS-Tag` der einzelnen Token betrachtet. Es kann beobachtet werden, dass häufig gewisse Wortart-Sequenzen bei Straßennamen auftauchen. Beispielsweise besteht der Straßenname „An der Eiche“ aus der Wortart-Sequenz (Präposition → Artikel → Substantiv). Auf dieser Grundlage wurden sechs POS-Tag-Muster definiert (siehe Tabelle 39 in Anhang A.4). Wenn die POS-Tags einer Tokensequenz eines dieser Muster erfüllen, wird die Tokensequenz als Straßennamenkandidat angenommen. Hierzu werden `StraßeAnfang` und `StraßeEnde` für den jeweiligen Anfang beziehungsweise das Ende der Tokensequenz auf `True` gesetzt und `StraßePOSMuster` für alle Token der Sequenz auf `True` gesetzt. Die gewählten POS-Tag Muster sind teilweise relativ generisch, so dass davon auszugehen ist, dass auch viele Straßennamenkandidaten erkannt werden, die keine Straßennamen sind. Durch die Betrachtung der Abhängigkeiten während der Aggregation, sollen jedoch diese falschen Treffer wieder entfernt werden.

Falls der Term „Postfach“ auftaucht, wird auch dieser als Kandidat für das Attribut *Straßenname* behandelt, um in der späteren Aggregation überprüfen zu können ob es einen Kandidaten für eine Postfachnummer in der textuellen Umgebung gibt. Dazu wird `StraßePostfach` auf `True` gesetzt.

7.2.2.5 *Identifikation von Unternehmensnamen*

Ebenso wie die Attribute *Städtename* und *Straßenname* unterliegen die Attributwerte des Attributs *Unternehmensname* keiner festen internen Struktur. Sie bestehen aus einer Tokensequenz flexibler Länge. Viele Unternehmen tragen jedoch ihre Rechtsform im Namen. Zur Identifikation von Kandidaten für das Attribut *Unternehmensname* werden zwei verschiedene Ansätze genutzt, von denen der erste das häufige Auftauchen der Rechtsform nutzt.

Durch Zugriff auf die Wissensbasis *Wikipedia*, im spezifischen die Liste internationaler Rechtsformen[11], wurde für den ersten Ansatz zunächst eine Liste an Namen und Abkürzungen für deutsche Rechtsformen erstellt. Hierbei wird unterschiedliche Kapitalisierung nicht beachtet, alle Einträge sind in Kleinbuchstaben enthalten. Diese Liste enthält 22 Einträge, unter Berücksichtigung verschiedener Schreibweisen sind 28 Einträge enthalten (siehe Tabelle 40 in Anhang A.4). Im Folgenden wird jede Tokensequenz welche unter Nichtbeachtung der Kapitalisierung mit einem Element dieser Indikatorenliste übereinstimmt als Teil eines Unternehmensnamenkandidaten markiert. Da ein Unternehmen in der Regel nicht nur aus dem Bezeichner einer Rechtsform besteht, müssen weiterhin Anfang und Ende des Unternehmensnamen identifiziert werden. Dafür werden ausgehend vom bereits bekannten Teil des Unternehmensnamenkandidaten (der Rechtsform beziehungsweise ihrer Abkürzung) sukzessive Token auf gewisse Merkmale überprüft. Ein Token wird dann als An-

11 http://de.wikipedia.org/wiki/Rechtsform, letzter Zugriff am 06.08.2015

fang oder Ende eines Unternehmensnamenkandidaten angenommen, wenn es am Zeilenanfang oder -ende steht, Teil eines anderen Attributskandidaten ist oder der Abstand zwischen aktuellem Anfang und Ende die maximale Distanz von elf Token überschreitet. Bei den Token der Sequenz wird `UnternehmenPerIndikator=True` gesetzt.

Um auch Unternehmensnamen identifizieren zu können, die nicht die Rechtsform des Unternehmens enthalten, wird die Tokensequenz, die einem Straßennamenkandidaten vorausgeht, als Unternehmensnamenkandidat angenommen. Dieser Schritt wird nur vorgenommen, falls kein Unternehmensnamenkandidat per Indikator gefunden wurde. Ebenso wie bei der ersten Heuristik wird ein Token als der Anfang der Tokensequenz angenommen, wenn es am Zeilenanfang steht, Teil eines anderen Attributskandidaten ist oder die maximale Länge der Tokensequenz von elf Token erreicht wurde. Bei den Token der Sequenz wird `UnternehmenPerStruktur=True` gesetzt.

Für beide Ansätze werden bei Anfang und Ende der Tokensequenz das Merkmal `UnternehmensAnfang` beziehungsweise das Merkmal `UnternehmensEnde` mit `True` belegt.

7.2.3 *Aggregation zu kompletten Adressen*

Nachdem nun die Kandidaten für die atomaren Attribute bestimmt wurden, werden diese im folgenden Schritt zu aggregierten Attributen zusammengefasst. Hierzu werden die Kandidaten für das Attribut *Unternehmensname* als Ausgangspunkt verwendet. Beginnend beim ersten Kandidaten sollen für jeden der Unternehmensnamenkandidaten die weiteren notwendigen atomaren Attribute, also *Straßenname, Hausnummer, Postleitzahl* und *Städtename*, identifiziert werden. Hierzu muss für jedes Attribut unter den identifizierten Kandidaten derjenige Kandidat gefunden werden, der zum aktuell betrachteten Unternehmensnamenkandidaten zugehörig ist.

Hierzu wird zunächst die Kombination der Attributskandidaten *Straßenname* und *Hausnummer* gesucht, zu der der geringstmögliche Tokenabstand liegt. Falls für die Token des Unternehmensnamen das Merkmal `UnternehmenPerIndikator` auf `True` gesetzt ist, wird innerhalb der nächsten zehn Zeilen gesucht (durch Zählen der Anzahl an Token mit `Zeilenende=True`). Falls `UnternehmenPerStruktur=True` gilt, wird nur in den folgenden beiden Zeilen gesucht, da hier das Risiko falscher Unternehmensnamenkandidaten höher ist. Falls ein Token mit `UnternehmensAnfang=True` zwischen dem aktuellen Kandidaten für das Attribut *Unternehmensname* und dem Kandidaten für das Attribut *Straßenname* liegt, wird der aktuelle Unternehmensnamenkandidat verworfen und mit dem neuen Unternehmensnamenkandidat fortgefahren, da dieser näher am Straßennamen steht. Die Kombination aus Token mit `StraßePostfach=True` und `Postfachnummer=True` wird identisch wie eine Straße mit Hausnummer behandelt. Wenn keine Kombination aus Straßennamen und Hausnummer in den gegebenen textuellen Abständen gefunden werden kann, wird direkt mit dem nächsten Schritt fortgefahren, da in Deutschland große Unternehmen eine eigene Postleitzahl haben können und in diesem Fall kein Attribut *Straßenname* zur Adresse gehört.

Im Folgenden wird die nächste Kombination aus Kandidaten für das Attribut *Postleitzahl* und das Attribut *Städtename* dem Attribut *Unternehmensname* zugeordnet.

Hierzu werden die folgenden fünf Zeilen durchsucht. Eine größere Zeilenanzahl könnte zu vielen Fehlzuordnungen führen. Auch hier wird der aktuelle Unternehmensnamenkandidat verworfen, wenn zwischen dem aktuellen Kandidaten und der folgenden Kombination von Kandidaten für die Attribute *Postleitzahl* und *Städtename* ein weiterer Kandidat für das Attribut *Unternehmensname* auftaucht.

Eine vollständige Adresse wird angenommen, wenn alle notwendigen atomaren Attribute vorliegen. Dies sind *Unternehmensname*, *Straßenname* oder das Token „Postfach", *Hausnummer*, welche auch eine Postfachnummer sein kann, *Postleitzahl* und *Städtename*. Da wie oben beschrieben, große Unternehmen nicht notwendigerweise einen Straßennamen mit Hausnummer benötigten, werden auch Adressen ohne Straßenname und Hausnummer als valide angenommen. Das beschriebene Vorgehen wird mit dem nächsten Kandidaten für das Attribut *Unternehmensname* wiederholt, bis alle Kandidaten für dieses Attribut einmal betrachtet wurden.

7.3 EVALUATION DES VERFAHRENS

Das vorgestellte Verfahren zur Identifikation postalischer Adressen wird in der Domäne *Impressumsseiten* evaluiert (siehe Abschnitt 4.2.2). Im Folgenden wird zunächst auf den zur Evaluation verwendeten Korpus eingegangen, weiterhin werden technische Details der zur Evaluation verwendeten Implementierung und dem Evaluationsvorgehen beschrieben. Nach Präsentation der Evaluationsergebnisse erfolgt eine Diskussion der Ergebnisse mit einem Vergleich zu anderen Ansätzen, die das Ziel der Identifizierung von Adressen verfolgen.

7.3.1 *Evaluationsdaten*

Aufgrund der in Deutschland herrschenden Impressumspflicht [24] muss jede Webseite eine Impressumsseite besitzen, die wiederum die postalische Anschrift des Besitzers der Webseite enthält. Zur Evaluation wird ein Korpus bestehend aus deutschsprachigen Impressumsseiten von deutschen Unternehmenswebseiten verwendet. Bei diesen Seiten kann davon ausgegangen werden, dass sie jeweils mindestens die geforderte Adresse enthalten. Der Korpus besteht aus 1.576 Dokumenten, welche bereits vom HTML-Format in ein Rohtextformat überführt wurden. Annotiert sind die Dokumente im Korpus mit der enthaltenen Adresse des Webseitenbesitzers, also der Unternehmensadresse. Dieser annotierte Korpus wird als Goldstandard verwendet, so dass die automatisiert identifizierten Adressen gegen die Annotationen abgeglichen werden können. Das Annotationsformat ermöglicht einen Abgleich auf Basis der atomaren Attribute. Dies bedeutet, dass für jede Adresse klar ersichtlich ist, welches atomare Attribut welchen Attributwert haben sollte. Einige der Impressumsseiten enthalten neben der Adresse des Webseitenbesitzers noch weitere Adressen, beispielsweise die des Webseitenerstellers oder alternative Adressen des Webseitenbesitzers. Diese weiteren Adressen sind nicht im Goldstandard enthalten, werden jedoch während der Evaluation bei Berechnung der Precision berücksichtigt. Bei genauerer Betrachtung der Impressumsseiten kann beobachtet werden, dass häufig die Adresse nicht im vorgesehenen Format gemäß der entsprechenden DIN-Norm [118] vorliegt. So findet sich der Name eines Unternehmens in zahlreichen Fällen nicht in

der textuellen Umgebung der eigentlich Adresse oder es befinden sich Token, die nicht zur Adresse gehören, zwischen den atomaren Attributen der Adresse.

7.3.2 *Evaluationsmethodik*

Zur Evaluation des Verfahrens wurde das Konzept unter Verwendung von Java 1.6 implementiert. Während der Vorverarbeitung wurden Methoden der Klasse `opennlp.tools.tokenize.TokenizerME` aus dem Framework *Apache OpenNLP* [83] zur Tokenisierung verwendet. Weiterhin wurde der *TreeTagger* [146] zur POS-Annotation verwendet.

Der Zugriff auf die benötigten OSM-Daten erfolgte per *Overpass API*[12]. Um die benötigte Liste deutscher Straßennamen zu erhalten, wurden alle Elemente der OSM-Datentypen `residential`, `living_street` und `pedestrian` in Deutschland abgerufen. Zur Erzeugung der Liste deutscher Städtenamen wurden die Elemente der OSM-Datentypen `city`, `town`, `suburb` und `village` genutzt.

Als Evaluationsmaße werden Precision, Recall und F1-Maß verwendet (siehe Abschnitt 2.2.3.2). Diese Maße werden sowohl für die einzelnen Elemente einer Adresse getrennt als auch für die komplette Adresse bestimmt. Es werden folgende Recall-Werte bestimmt:

- r_{Gesamt}: der Anteil der Unternehmensadressen im Goldstandard, die vollständig korrekt identifiziert wurden; eine Adresse ist genau dann vollständig korrekt, wenn alle atomaren Attribute korrekt identifiziert wurden,
- $r_{\text{Unternehmen}}$: der Anteil der Menge der Attributwerte für das Attribut *Unternehmensname* im Goldstandard, die korrekt identifiziert wurden,
- $r_{\text{StraßeUndNummer}}$: der Anteil der Kombinationen aus den Attributen *Straßename* und *Hausnummer* im Goldstandard, die korrekt identifiziert wurden,
- $r_{\text{PLZUndStadt}}$: der Anteil der Kombinationen aus den Attributen *Postleitzahl* und *Städtename* im Goldstandard, die korrekt identifiziert wurden.

Weiterhin soll untersucht werden, welcher Anteil der Adressen ohne Berücksichtigung des Attributs *Unternehmensname* korrekt identifiziert wurde. Dafür werden während der Aggregation nicht die Unternehmensnamenkandidaten, sondern die Kombination aus Straße und Hausnummer als Ausgangspunkt verwendet. Daraus ergibt sich r_{Geodaten}, welches dem Anteil der Adressen aus dem Goldstandard entspricht, der unter Nichtbeachtung des Attributs *Unternehmensname* vollständig korrekt identifiziert wurde.

Da der Goldstandard nur die Adressen der Webseitenbesitzer in annotierter Form enthält und weitere Adressen auf den Impressumsseiten nicht annotiert sind, ist eine automatisierte Evaluation der Precision nicht möglich. Das vorgestellte Verfahren identifiziert mitunter Adressen, die nicht im Goldstandard enthalten sind, aber dennoch korrekt sind. Zur Evaluation der Precision wurden die identifizierten Adressen der ersten 100 Impressumsseiten herangezogen und diese manuell auf ihre Korrektheit überprüft. Die einzelnen Precision-Maße p_{Gesamt}, $p_{\text{Unternehmen}}$, $p_{\text{StraßeUndNummer}}$,

12 http://overpass-api.de/, letzter Zugriff am 15.08.2015

$p_{PLZUndStadt}$ und $p_{Geodaten}$ sind analog zu den Recall-Maßen definiert. Gleiches gilt für die F1-Maße.

Ein einzelnes Attribut wird genau dann als korrekt angenommen, wenn der identifizierte Attributwert vollständig mit dem im Goldstandard vorgegebenen Attributwert übereinstimmt. Dies bedeutet, dass beispielsweise ein unvollständig identifizierter Unternehmensname nicht als korrekt angenommen wird, selbst wenn der unvollständige Unternehmensname einem Menschen zur eindeutigen Identifikation eines Unternehmens genügen würde.

7.3.3 *Ergebnisse*

Insgesamt wurden mittels des vorgestellten Verfahrens im Evaluationskorpus 4.449 Adressen identifiziert, dies entspricht durchschnittlich 2,8 Adressen pro Dokument, also pro Impressumsseite. Wie schon zuvor erläutert, sind auf den Impressumsseiten nicht nur die Adressen der Webseitenbesitzer genannt, sondern auch andere Adressen, wie die des Webdesigners. Häufig taucht die Adresse des Seitenbesitzers mehrfach auf einer Seite auf. Insbesondere in der Fußzeile einer Webseite lassen sich in vielen Fällen die Kontaktdaten des Unternehmens finden.

Eine Übersicht über die Ergebnisse der Evaluation ist in Abbildung 22 dargestellt. Es zeigt Recall, Precision und F1-Maß für die Identifikation der kompletten Unternehmensadresse, für den Unternehmensnamen, für die reine Adresse ohne Unternehmensnamen (Geodaten), für die Straße mit Hausnummer sowie die Postleitzahl mit Stadt.

Die Betrachtung des Wertes für den Recall zeigt, dass ein großer Anteil der zu identifizierenden Adressen der Webseitenbesitzer korrekt identifiziert wurde (79,9%). Insbesondere hat hierbei jedoch die Identifikation des Unternehmensnamen einen dämpfenden Einfluss mit einem Recall-Wert von 82,2%. Für die reinen Geodaten fällt der Recall-Wert mit 94,7% deutlich besser aus. Die Identifikation des Tupels aus den Attributen *Straße* und *Hausnummer* funktioniert ähnlich gut wie die des Tupels aus *Postleitzahl* und *Stadt* (95,3% beziehungsweise 96,9%).

In Hinblick auf die Precision schneidet das vorgestellte Verfahren etwas schlechter ab. So sind nur 61,5% der identifizierten Attributwerte für das Attribut *Unternehmensadresse* vollständig korrekt. Auch hier stellt die Identifikation des Unternehmensnamen mit einem Precision-Wert von 62,6% einen dämpfenden Einfluss dar. Die Geodaten, sowie deren Komponenten werden jeweils mit einem relativ hohen Precision-Wert identifiziert (zwischen 92,0% und 93,1%).

Der bei Precision und Recall zu erkennende Trend, dass die Erkennung des Attributs *Unternehmensname* die größte Fehlerrate hat, lässt sich aufgrund dessen Definition auch im F1-Maß wiedererkennen. Hier liegt das Ergebnis für die Erkennung der kompletten Adresse und des Attributs *Unternehmensname* mit 69,5% beziehungsweise 71,1% deutlich unter der Erkennung der Geodaten (93,9%) sowie deren Komponenten *Straße* und *Hausnummer* (94,2%) sowie *Postleitzahl* und *Stadt* (94,4%).

Bei manueller Analyse der Ergebnisse wurde beobachtet, dass gewisse Terme häufig fälschlicherweise als Kandidaten für atomare Attribute identifiziert werden[13]. Dies führt folglich auch zu fehlerhaften Identifikationen des aggregierten Attributs. Aus diesem Grund wurden manuell Tabulisten erstellt. Wenn ein Element einer sol-

13 Dies sind Terme wie „Steueridentifikationsnummer“, die häufiger auf Impressumsseiten auftauchen.

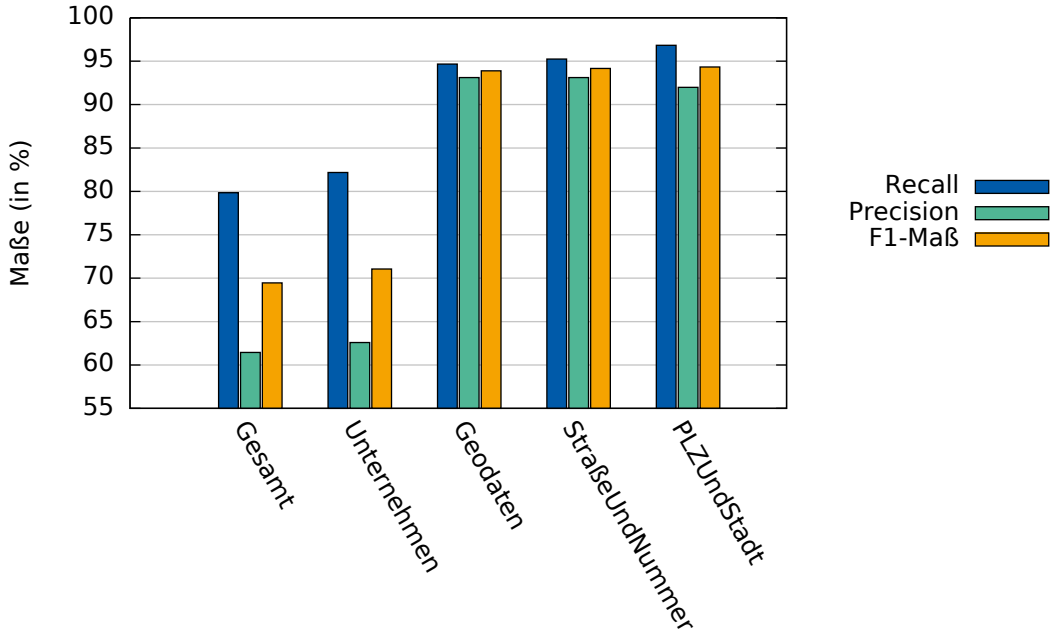


Abbildung 22: Ergebnisse der Identifikation von postalischen Unternehmensadressen

chen Tabuliste als Kandidat für ein atomares Attribut oder als dessen Teil identifiziert wird, so wird dieser Kandidat verworfen. Für die Identifikation der Kandidaten für das Attribute *Städtename* beinhaltet die Tabuliste sieben Einträge, für das Attribut *Straßenname* 19 Einträge und für das Attribut *Unternehmensname* 35 Einträge. Durch Verwendung dieser Tabulisten konnten die Ergebnisse weiter verbessert werden. Die Ergebnisse für die Identifikation der kompletten Adresse sowie nur der Geodaten ist Tabelle 24 zu entnehmen. Auffällig ist insbesondere die Verbesserung um etwa 4 Prozentpunkte für die einzelnen Maße für die Identifikation der kompletten Adresse. Demnach hat insbesondere die Tabuliste für Kandidaten für das Attribut *Unternehmensname* einen erheblichen Einfluss. Aber auch die Ergebnisse für die Geodaten verbessern sich um 0,3 Prozentpunkte (Precision) bis 1,0 Prozentpunkte (Recall) durch die Tabulisten für die Attribute *Straßenname* und *Städtename*.

Tabelle 24: Ergebnisse bei Verwendung von Tabulisten für die einzelnen Attributskandidaten

	Gesamt (regulär)	Gesamt (Tabuliste)	Geodaten (regulär)	Geodaten (Tabuliste)
Recall	79,9%	83,3%	94,7%	95,7%
Precision	61,5%	65,7%	93,1%	93,4%
F1-Maß	69,5%	73,4%	93,9%	94,5%

Weiterhin hat die manuelle Analyse der Ergebnisse gerade bei der Identifikation des Attributs *Unternehmensname* gezeigt, dass häufig Teile des Attributs korrekt bestimmt wurden, aber der Unternehmensname nicht vollständig korrekt war und somit nicht als korrekte Identifikation gewertet wurde. Dies hat sowohl einen Einfluss auf die Precision als auch auf den Recall und somit auch das F1-Maß. Weiterhin kam

es teilweise zu Fehlzuordnungen zwischen Unternehmensname und Geodaten. Dies resultiert insbesondere aus der ursprünglichen Erstellung der Dokumente auf Basis der Webseiten. Bei der Konvertierung von HTML-Dateien zu Rohtext kam es vor, dass beispielsweise bei Tabellen textuelle Fragmente nach Konvertierung unabhängig von ihrem eigentlichen textuellen Kontext auftauchten.

7.3.4 *Diskussion alternativer Ansätze*

Da bereits Verfahren existieren, die sich mit der Identfikation von postalischen Adressen beschäftigen, wird in diesem Abschnitt ein kurzer Überblick über die relevanten verwandten Ansätze gegeben. Neben der Beschreibung erfolgt abschließend eine Darstellung der Evaluationsergebnisse der einzelnen Ansätze mit einem Vergleich zu dem in dieser Arbeit vorgestellten Verfahren.

Von Loos und Biemann [93] wurde ein Ansatz vorgestellt, der Conditional Random Fields (CRF) zur Identifikation von Adressen nutzt. CRF [1] werden aufgrund ihrer Eignung zur Klassifikation sequentieller Daten verwendet (siehe Abschnitt 2.2.1). Im vorgestellten Ansatz wird sowohl ein annotierter Goldstandard-Datensatz aus 400 Dokumenten als auch ein weiterer unannotierter Datensatz verwendet. Zunächst werden die im unannotierten Datensatz enthaltenen Terme unter Verwendung eines unüberwachten Lernverfahrens geclustert, anschließend werden die resultierenden Cluster als Merkmale für die CRF verwendet. Diese CRF werden unter Verwendung des annotierten Datensatzes trainiert. Zur Evaluation wurden Webseiten von deutschen Restaurants, Geschäften und anderen Einrichtungen verwendet. Die Autoren konnten zeigen, dass die Verwendung des unannotierten Datensatzes einen deutlich positiven Einfluss auf die Klassifikationsgüte hat.

Im Gegensatz zu diesem statistischen Ansatz stehen regelbasierte Verfahren. Asadi et al. [6] verwenden manuell erstellte Regeln mit unterschiedlichen Gewichten. Diese Regeln werden durch Hinzunahme von Listen geographischer Daten, wie Namen großer Städte, angereichert. Es hat sich gezeigt, dass die Hinzunahme dieser Listen die korrekte Identifikationsquote erhöhen kann. Evaluiert wurde das Verfahren unter Verwendung australischer Webseiten. Cai et al. [26] stellen ebenso einen regelbasierten Ansatz vor, welcher durch Daten aus einem geographischen Informationssystem angereichert wird. Weiterhin werden ontologische Daten verwendet. Zur Extraktion findet ein graphenbasierter Vergleich zwischen den definierten Regeln und den Wortsequenzen im Text statt. Als Evaluationsdaten werden die Einträge eines Branchenverzeichnisses verwendet. Daher kann davon ausgegangen werden, dass sich die einzelnen Einträge im Evaluationskorpus in ihrer Vorstrukturierung stark ähneln. Das Verfahren *Ge(o)Lo(cator)* [117] identifiziert Adressen unter Verwendung von POS-Tags, linguistischen Regeln und Listen geographischer Entitäten. Neben der reinen Extraktion von Adressen steht die Anreicherung mittels Längen- und Breitengrade einer Adresse im Vordergrund. Evaluiert wurde das Verfahren unter Verwendung von Webseiten italienischer Universitäten, Forschungseinreichtungen und Unternehmen. Im Ansatz von Ahlers und Boll [3] werden nicht nur Listen von geographischen Entitäten eines Typs verwendet, es werden weiterhin Relationen zwischen den Entitäten betrachtet. So wird beispielsweise auf Basis der geographischen Informationssysteme validiert, ob eine spezifische Kombination aus Postleitzahl und Städtenamen gültig ist, also ob diese Kombination in Deutschland vorkommt. Die-

Tabelle 25: Vergleich mit verwandten Arbeiten (angegeben sind die jeweils in der Publikation genannten Evaluationsergebnisse)

	Precision (in %)	Recall (in %)	F1 (in %)	Ansatz	Listen-grösse	Land
[93][14]	89,1	63,5	74,1	statistisch	keine	DE
[6]	97	73	83	Regeln	klein	AUS
[26]	74,5	72,4	73,4	Regeln	groß	CAN
[117]	90,5	92,7	91,6	Regeln	groß	IT
[3]	k.A.	~ 95	k.A.	Regeln	groß	DE
diese Arbeit	93,1	94,7	93,9	Regeln	mittel	DE

ser Ansatz ermöglicht eine hohe Klassifikationsgüte, ist aber stark von der Aktualität der verwendeten geographischen Informationssysteme abhängig.

Bei einer Literatursuche konnte kein Ansatz identifiziert werden, der sich mit der Identifikation von Unternehmensadressen beschäftigt, was das Ziel des in dieser Arbeit vorgestellten Verfahrens ist. Aus diesem Grunde werden im Folgenden die verwandten Arbeiten mit der reinen Identifikation der Geodaten mittels des in dieser Dissertation neu vorgestellten Verfahrens verglichen.

Tabelle 25 gibt eine Übersicht über die verwandten Arbeiten und die erzielten Werte für Precision, Recall und F1-Maß. Im Vergleich dazu sind die Werte für das in dieser Arbeit vorgestellte Verfahren dargestellt. Gegeben sind die Werte für die Identifikation der Geodaten. Es ist zu beachten, dass die Evaluationen mit unterschiedlichen Datensätzen durchgeführt wurden und die Identifikation von Adressen unterschiedlicher Länder betrachtet wurde. Daher sind die Ergebnisse nicht direkt miteinander vergleichbar, sondern können nur eine Tendenz widerspiegeln. Weiterhin sind die Arbeiten in der Tabelle nach dem vorherrschenden Ansatz klassifiziert und eine Angabe der Größe der verwendeten Geodatenliste sowie das fokussierte Land sind gegeben.

Insgesamt schneidet der in dieser Arbeit vorgestellte Ansatz besser als die verwandten Ansätze ab. Zwar erreicht der Ansatz von Asadi et al. [6] eine höhere Precision, aber der Recall-Wert fällt geringer aus, was sich auch auf einen insgesamt niedrigeren Wert für das F1-Maß auswirkt. Weiterhin hat der Ansatz von Ahlers et al. [3] einen höheren Recall-Wert, dieser basiert jedoch nur auf einer Schätzung und Werte für Precision und F1-Maß sind nicht gegeben.

7.4 FAZIT

In diesem Kapitel wurde ein Verfahren zur Identifikation aggregierter Attribute vorgestellt. Zunächst wurde dabei auf ein domänen- und attributsunabhängiges Konzept eingegangen. Dieses nimmt unter Verwendung globaler Regeln für das

14 Die angegebenen Werte für diesen Ansatz stellen die durchschnittlichen Werte für die Identifikation der einzelnen atomaren Attribute dar. Bei allen anderen Ansätzen ist der Wert für die Identifikation der kompletten Geodaten gegeben.

betrachtete aggregierte Attribut und lokaler Regeln für die Identifikation der atomaren Attribute die Identifikation vor. Unter Betrachtung des aggregierten Attributs deutscher Unternehmensadressen wurde das vorgestellte Verfahren konkretisiert. Zur Identifikation der atomaren Attribute wurden Daten der frei verfügbaren OpenStreetMap (OSM)-Datenbasis verwendet. Da OSM eine sehr gute Abdeckung für zahlreiche Länder aufweist, kann diese Datenbasis auch zur Identifikation von Adressen in anderen Ländern verwendet werden. Der zugrunde liegende regelbasierte Ansatz erlaubt eine gute Übertragbarkeit auf Adressen anderer Länder. Hierzu müssen einige landesspezifische Charakteristika von Adressen definiert werden. In einer Evaluation konnte gezeigt werden, dass die Güte des Verfahrens (F1-Maß für die Identifikation bei 93,9%) die Güte anderer Verfahren mit der Zielsetzung der Identifikation postalischer Adressen (F1-Maß von 73,4% bis maximal 91,6%) übersteigt. Somit eignet sich das Verfahren sehr gut zur Identifikation von Adressen. Weiterhin ist es übertragbar auf andere aggregierte Attribute.

8

KOMBINIERTER ANSATZ ZUR STRUKTURIERUNG

In den vorigen Kapiteln dieser Dissertation wurden Verfahren zur Identifikation einzelner Attribute in Dokumenten vorgestellt. Da in dieser Arbeit untersucht werden soll, mit welchen Methoden eine Strukturierung von Dokumenten durchgeführt werden kann, wird ebenso ein Konzept benötigt, welches die gemeinsame Nutzung von Methoden zur Identifikation ermöglicht, um das Ziel der Dokumentenstrukturierung zu erreichen.

Wie sich gezeigt hat, kommt es bei Verfahren zur Identifikation einzelner Attribute, wie zum Beispiel bei den in Kapitel 5, Kapitel 6 und Kapitel 7 vorgestellten Verfahren oder auch den Verfahren zur Identifikation von Eigennamen aus verwandten Arbeiten (vergleiche Kapitel 3.3.1), zu fehlerhaften Erkennungen. Die Fehlerrate ist stark vom jeweiligen Attribut abhängig. Während beispielsweise postalische Adressen mit einer relativ hohen Güte identifiziert werden können (siehe Abschnitt 7.3.3), gibt es bei der Identifikation von Freitextattributen eine größere Fehlerrate (siehe Abschnitt 5.2.3). Bei der Kombination der Verfahren sollte dies berücksichtigt werden, um das Ziel einer möglichst geringen Fehlerrate für die Strukturierung insgesamt zu erreichen.

Weiterhin sollte eine spezifische im Dokument enthaltene Tokensequenz nicht gleichzeitig mehreren Attributen eines Objektes zugeordnet werden. Dies erfordert den Ausschluss einer Tokensequenz bei der Identifikation weiterer Attribute, wenn diese Tokensequenz bereits einem Attribut zugeordnet wurde. Außerdem ist zu beachten, dass Attributwerte unterschiedliche Umfänge haben und ihr Vorkommen im Text in unterschiedlichen textuellen Umgebungen zu beobachten ist. So kann beispielsweise davon ausgegangen werden, dass in der Zeile eines Dokumentes, in welcher das Attribut *Titel* einer Ausschreibung für eine studentische Abschlussarbeit zu finden ist, keine weiteren Attributwerte zu finden sind. Dies sollte ebenso beim Konzipieren eines kombinierten Verfahrens berücksichtigt werden.

In diesem Kapitel wird ein Konzept zur Kombination von Verfahren zur Identifikation einzelner Attribute zunächst generisch vorgestellt. Im Weiteren wird das Konzept mittels einer Fallstudie in der Domäne der *Ausschreibungen studentischer Abschlussarbeiten* (siehe Abschnitt 4.2.3) konkretisiert und eine Umsetzung in dieser Domäne evaluiert [61]. Die Evaluation dient der Überprüfung, ob unabhängige Verfahren zur Identifikation von Attributen gemeinsam verwendet werden können, um das Ziel der Strukturierung von Dokumenten zu erreichen.

8.1 KONZEPT

Ähnlich wie bei der Identifikation aggregierter Attribute (Kapitel 7) wird ein Ansatz angewendet, bei dem über die Identifikation der einzelnen zu bestimmenden Attribute im Dokument iteriert wird. Bei dem hier vorgestellten Ansatz können jedoch in einem einzelnen Schritt mehrere Attribute identifiziert werden, dies trifft

bei den Freitextattributen zu. Eine Menge von in einem Schritt zu identifizierenden Attributen wird als *Attributsgruppe* bezeichnet. Zur Identifikation von Attributen einer Gruppe von Freitextattributen wird ein multinominaler Klassifikator verwendet. Dieser führt in einem Schritt die Klassifikation eines Textsegmentes in eines der möglichen Freitextattribute aus, somit wird die Identifikation aller Freitextattribute, die mittels Klassifikation identifiziert werden, in einem Schritt durchgeführt.

Das vorgestellte Konzept ist in Abbildung 23 skizziert. Zunächst erfolgt eine Gruppierung von Attributen in Attributsgruppen (1). Während bei aggregierten Attributen von einer festen Reihenfolge der atomaren Attribute ausgegangen werden kann, trifft dies aufgrund der heterogenen Struktur der Dokumente für die Identifikation der Gesamtmenge der Attribute nicht zu. Die Reihenfolge der einzelnen Iterationsschritte orientiert sich stattdessen an der Güte der Verfahren zur Identifikation der einzelnen Attributsgruppen (2). Insgesamt sollte die Sortierung so vorgenommen werden, dass eine Maximierung der Güte des kombinierten Verfahrens erreicht wird. Daher wird zunächst die Attributsgruppe identifiziert, von deren Identifikation die höchste Güte erwartet wird (3,4). Dies sind in der Regel Attributsgruppen mit homogener interner Struktur, die sich durch regelbasierte Ansätze identifizieren lassen. Sobald eine Attributsgruppe erfolgreich identifiziert wurde, wird deren Werte, also beispielsweise eine kompletten Textzeilen, aus dem Text entfernt, da die entsprechende Textsequenz keinem weiteren Attribut mehr zugeordnet werden kann (5). Dies gilt nicht für Meta-Attribute, da sie nicht wörtlich im Text zu finden sind. Falls vorhanden, sollten Meta-Attribute als erstes identifiziert werden, da sie sich auf den kompletten Text beziehen und das Entfernen von Textsegmenten deren Identifikation somit verschlechtern kann. Das Verfahren wird wiederholt (7), bis die Attribute aller Attributsgruppen identifiziert wurden. Sobald alle Attribute identifiziert wurden, wird die Menge der Attributwerte zurückgegeben (8).

8.2 FALLSTUDIE: AUSSCHREIBUNGEN STUDENTISCHER ABSCHLUSSARBEITEN

Zur Untersuchung der Nutzbarkeit wird das im vorigen Abschnitt vorgestellte Konzept in einer Fallstudie für eine konkrete Domäne und eine Entitätsausprägung ausgestaltet. Dazu wird die Domäne *Ausschreibungen studentischer Abschlussarbeiten* gewählt. Diese Domäne weist mit der vorgestellten Entitätsausprägung (Kapitel 4.2.3) eine Heterogenität der Attributtypen auf und eignet sich daher zur Evaluation des kombinierten Ansatzes mittels verschiedener Identifikationsansätze.

Zunächst wurde ein kleiner Beispieldatensatz von 25 Dokumenten aus dieser Domäne manuell auf das Vorkommen der in Abschnitt 4.2.3 beschriebenen Attribute in Ausschreibungen studentischer Abschlussarbeiten analysiert [176]. Der Anteil der Dokumente, in denen die einzelnen Attribute vorkommen, ist in Tabelle 26 dargestellt.

Zur Umsetzung des Strukturierungsverfahrens findet eine Fokussierung auf die Attribute statt, die in mindestens 50% der Dokumente vorhanden sind. Eine Übersicht über die einzelnen Attributsgruppen, die darin betrachteten Attribute sowie die jeweils zur Identifikation gewählten Ansätze ist Tabelle 27 zu entnehmen. In dieser Entitätsausprägung werden zunächst drei Attributsgruppen mit nur einem Attribut verwendet und nachfolgend eine weitere Gruppe mit den drei verbleibenden Attributen. Die Werte der im Folgenden verwendeten Parameter für maximale

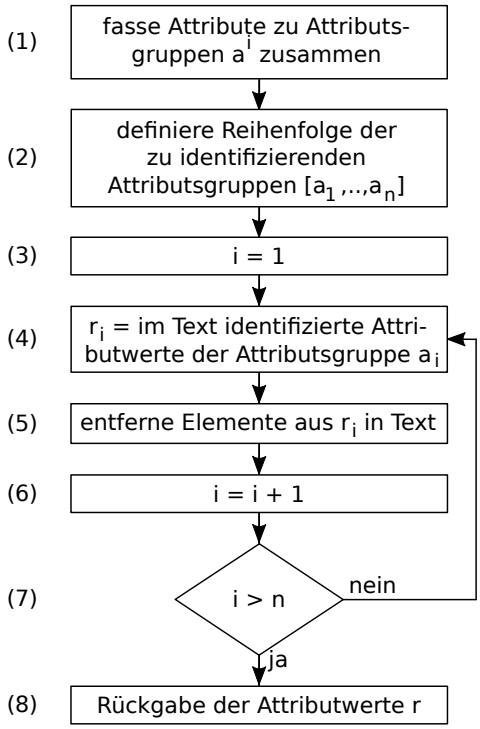


Abbildung 23: Darstellung des Konzepts zum kombinierten Ansatz zur Strukturierung textueller Dokumente

Zeilen- oder Absatzlängen wurde auf Basis einer Analyse von Beispieldokumenten festgelegt. [61]

Die Identifikation des Attributs *E-Mail-Adresse* erfolgt mittels eines regulären Ausdrucks. Bei diesem regelbasierten Ansatz ist aufgrund der einheitlichen internen Struktur von E-Mail-Adressen eine geringe Fehlerrate anzunehmen, weshalb die Identifikation des Attributs *E-Mail-Adresse* im ersten Schritt durchgeführt wird. Anschließend wird die Textzeile, in der das Attribut *E-Mail-Adresse* auftaucht, entfernt, da in der gleichen Zeile kein weiterer Attributwert zu erwarten ist.

Ebenso regelbasiert erfolgt die Identifikation des Attributs *Typ der Arbeit*. Hierbei wird im Text nach Vorkommen einer der Zeichensequenzen „Master“, „Bachelor“, „Diplom“ oder „Studienarbeit“ in Groß- sowie Kleinschreibung gesucht. Die Textzeile, die die Zeichensequenz enthält, wird nur gelöscht, wenn sie kürzer als 25 Zeichen ist, da dann davon ausgegangen wird, dass sie kein Element des Fließtextes ist und somit kein weiteres Attribut enthält.

Zur Identifikation des Attributs *Titel* wird eine Heuristik angewendet. Obwohl das Attribut *Titel* ein Freitextattribut darstellt, ist das vorgestellte Verfahren zur Identifikation von Freitextattributen (Kapitel 5) hier nicht geeignet. Die im Titel vorliegende Wortverteilung entspricht dem Thema der Arbeit und es lässt sich keine für Titel im allgemeinen repräsentative Wortverteilung beobachten, welche zur Identifikation verwendet werden könnte. Stattdessen basiert die verwendete Heuristik auf der Annahme, dass Wörter des Titels häufig auch in der restlichen Ausschreibung auftauchen. Als Attributskandidaten werden die ersten drei Absätze des Textes betrachtet, da der Titel in der Regel eher am Anfang der Ausschreibung zu erwarten ist. Für jeden dieser Absätze wird analysiert, wie hoch die Überlappung der enthaltenen Wörter zum Rest des Textes ist. Der Absatz mit dem höchsten Grad der Überlappung wird

Tabelle 26: Häufigkeit einzelner Attribute in Ausschreibungen studentischer Abschlussarbeiten (nach [176])

Attribut	Häufigkeit	Attribut	Häufigkeit
Titel	100%	*Vorkenntnisse/ Voraussetzungen*	72%
Beschreibung	100%	*Institutsbeschreibung*	24%
Ziele	100%	*Veröffentlichungs-datum*	16%
E-Mail-Adresse	100%	*Sprache*	10%
Typ der Arbeit	92%	*Beginn*	8%

Tabelle 27: Betrachtete Attribute in der Domäne *Strukturierung von Ausschreibungen studentischer Abschlussarbeiten*, der gewählte Ansatz zur Identifikation sowie der Index der jeweiligen Attributsgruppe

Attributs-gruppe	Attribut	Beschreibung	Ansatz
1	*E-Mail-Adresse*	E-Mail-Adresse des Betreuers der Arbeit	regelbasierte Identifikation
2	*Typ der Arbeit*	Masterarbeit/ Bachelorarbeit/ Diplomarbeit/ Studienarbeit	regelbasierte Identifikation
3	*Titel*	Titel der Ausschreibung	Heuristik
4	*Beschreibung*	ausführliche Beschreibung der Aufgabenstellung	Identifikation von Freitext-attributen
4	*Ziele*	Ziele der Arbeit	Identifikation von Freitext-attributen
4	*Vorkenntnisse/ Voraussetzungen*	Voraussetzungen, die der Interessent mitbringen sollte	Identifikation von Freitext-attributen

als Attributwert für das Attribut *Titel* angenommen und aus dem Text entfernt. Der Grad der Überlappung eines Absatzes zum restlichen Text definiert sich über die Anzahl der Wörter der Absatzes, die im restlichen Text auftauchen, normiert über die Länge des Absatzes.

Abschließend erfolgt die Identifikation einer Gruppe von Attributen, nämlich der Attribute *Beschreibung*, *Ziele* und *Vorkenntnisse/Voraussetzungen* mittels der in Kapitel 5 vorgestellten Identifikation der Freitextattribute. Dafür wird der Text zunächst auf Basis einer Absatzerkennung segmentiert. Eine Segmentgrenze wird angenommen, wenn im Rohtext mindestens eine Leerzeile auftaucht. Anschließend werden die Seg-

mente, die mindestens 180 Zeichen enthalten, klassifiziert. Die einzelnen Segmente werden jeweils dem Attribut zugeordnet, für das der Klassifikator den höchsten Konfidenzwert anzeigt. Falls mehrere Segmente dem gleichen Attribut zugeordnet werden, werden die Segmente in konkatenierter Form als Attributwert angenommen. Zur Klassifikation wird nur das Merkmal der Unigramme, also die im Text vorkommenden Wörter, verwendet.

8.3 EVALUATION

Die prototypische Umsetzung der Fallstudie wird evaluiert in der Domäne *Ausschreibungen studentischer Abschlussarbeiten* (Abschnitt 4.2.3). Hierzu wurde zunächst ein Evaluationskorpus aufgebaut und dieser dann systematisch zur Evaluation verwendet. Die einzelnen Aspekte der Evaluation werden im Folgenden erläutert.

8.3.1 *Evaluationsdaten*

Für die Evaluation wurde ein Korpus aus 118 Ausschreibungen studentischer Abschlussarbeiten zusammengestellt. Dafür wurden manuell Webseiten von 17 verschiedenen deutschen Universitäten besucht und online verfügbare Ausschreibungen für studentische Abschlussarbeiten im PDF-Format gesammelt. Es wurden ausschließlich Ausschreibungen in deutscher Sprache gesammelt. Bei der Suche wurde nicht umfassend vorgegangen, da dies einen erheblichen manuellen Aufwand bedeutet hätte.

Die Ausschreibungen wurden mittels des Tools *PDFMiner*[1] vom PDF-Format in Rohtext überführt. Anschließend wurden alle zu identifizierenden Elemente im Rohtext manuell von einem Annotator annotiert. Diese manuellen Annotationen stellen den Goldstandard für die Evaluation dar. Die Häufigkeit der einzelnen Attribute in den Evaluationsdaten ist in Tabelle 28 dargestellt. Es ist zu beachten, dass Attribute auch mehrfach in einem Dokument auftreten können, dies betrifft insbesondere die Freitextattribute, da deren Werte teils fragmentiert im Dokument auftauchen. Außerdem tauchen teils mehrere E-Mail-Adressen auf, beispielsweise des betreuenden wissenschaftlichen Mitarbeiters und des Lehrstuhlinhabers.

Tabelle 28: Häufigkeit der einzelnen Attribute im Goldstandard (118 Dokumente insgesamt)

ATTRIBUT	ANZAHL
Titel	117
Typ der Arbeit	112
Beschreibung	139
Ziele	164
Vorkenntnisse/Voraussetzungen	89
E-Mail-Adresse	133

1 Verfügbar unter http://www.unixuser.org/~euske/python/pdfminer/, letzter Zugriff am 09.09.2015

8.3.2 *Evaluationsmethodik*

Zur Evaluation wurde eine k-fache Kreuzvalidierung durchgeführt. Aufgrund der kleinen Größe des Evaluationskorpus wurde $k = 5$ gewählt. Dies bedeutet, dass für jeden der k Evaluationsläufe 94-95 Dokumente als Trainingsdokumente verwendet werden und 23-24 Dokumente als Testdokumente. Eine größere Zahl k würde zu einer kleineren Zahl an Testdokumenten in den einzelnen Durchläufen führen. Ausgewertet wurden während der Evaluationen Precision und Recall für die einzelnen Attribute. Ein identifizierter Attributwert wurde dann als korrekt angenommen, wenn es nach Entfernen von äußeren Leerzeichen und Leerzeilen mit dem Attributwert des Goldstandards übereinstimmt.

Die Trainingsdaten in jedem Evaluationsdurchlauf werden verwendet, um den Klassifikator für die Freitextattribute *Beschreibung*, *Ziele* und *Vorkenntnisse/Voraussetzungen* zu trainieren. Da die verbleibenden drei Attribute *Titel*, *Typ der Arbeit* und *E-Mail-Adresse* mittels Heuristiken identifiziert werden, werden deren Trainingsdaten nicht benötigt.

8.3.3 *Ergebnisse*

Die Ergebnisse der wie beschrieben durchgeführten Evaluation sind in Tabelle 29 dargestellt. Gegeben sind die Ergebnisse für Precision und Recall für die einzelnen Attribute als arithmetisches Mittel über die fünf Evaluationsdurchläufe sowie das auf Basis der arithmetischen Mittel bestimmte F1-Maß.

Tabelle 29: Ergebnisse des kombinierten Verfahrens (in %)

Extrahiertes Attribut	Recall	Precision	F1
E-Mail	95	99	97
Typ der Arbeit	78	83	80
Titel	28	27	28
Ziele	65	56	60
Beschreibung	96	41	57
Voraussetzungen	25	61	35

Die besten Ergebnisse wurden für das Attribut *E-Mail-Adresse* erzielt. Eine qualitative Analyse der falsch erkannten Attributwerte für dieses Attribut zeigte, dass die nicht oder falsch erkannten E-Mail-Adressen im Text auf Fehler in der Zeichenkodierung zurückzuführen sind. Die fehlerhaften Zeichenkodierungen entstand bei der Konvertierung der PDF-Dokumente zu Rohtext. Etwas schlechter schneidet die Erkennung für das Attribut *Typ der Arbeit* ab. Die fehlerhaften Erkennungen lassen sich auf zwei Gründe zurückführen. Zum einen sind Inkonsistenzen in den Ausschreibungen zu finden. So wird explizit erwähnt, für welchen Typ von Abschlussarbeit eine Ausschreibung vorgesehen ist, aber im Text wird ein anderer Typ genannt. Zum anderen ist eine größere Variation der Bezeichner für die unterschiedlichen Typen von Abschlussarbeiten zu finden. Durch einen Einbezug der externen Struktur könnte auch die Erkennung von Abkürzungen wie „Ba“ oder „M“ für eine Verbesserung

des Recall-Wertes sorgen. Am schlechtesten schneidet die Erkennung des Attributs *Titel* ab. Bei Analyse der Fehler hat sich gezeigt, dass der Titel der Ausschreibung häufig in einer anderen Sprache verfasst ist als der Text. Aufgrund dessen gibt es keine Überschneidungen zwischen den im Titel und den in weiteren Attributen der Ausschreibung verwendeten Wörtern. Weiterhin hat sich gezeigt, dass häufig im Attribut *Titel* eine unterschiedliche Terminologie als in den anderen Attributen verwendet wird, dies trifft insbesondere bei provokanten, wenig technischen Titeln zu. Bei der Erkennung der Freitextattribute *Beschreibung* und *Ziele* wird jeweils ein F1-Maß von knapp 60% erreicht. Deutlich schlechter schneidet das Freitextattribut *Voraussetzungen* ab. Insbesondere fällt hierbei der schlechte Recall-Wert von 25% auf. Bei der Analyse der Ausschreibungen und der resultierenden strukturierten Daten konnte beobachtet werden, dass die Voraussetzungen für eine Arbeit häufig sehr kurz formuliert sind und weniger als die angenommenen 180 Zeichen enthalten.

Zur besseren Bewertung der beschriebenen Evaluationsergebnisse für die Freitextattribute wurde in dieser Domäne zusätzlich eine Klassifikation mittels des in Kapitel 5 vorgestellten Verfahrens zur Identifikation von Freitextattributen vorgenommen und mittels der in Abschnitt 5.2.2 vorgestellten Evaluation evaluiert. Der Klassifikator (SVM) wurde mit den Trainingsdaten trainiert und die einzelnen Attributwerte für die Freitextattribute wurden anschließend klassifiziert. Somit werden also nur Textfragmente, die einem der drei Attribute zuzuordnen sind, klassifiziert. Zur Evaluation wurde eine 10-fach stratifizierte Kreuzvalidierung genutzt. Es wurde nur das Merkmal der Unigramme zur Klassifikation verwendet. Die Ergebnisse sind in Tabelle 30 dargestellt. Für alle drei Attribute werden deutlich bessere Ergebnisse erzielt als innerhalb des kombinierten Ansatzes (vergleiche Tabelle 29). Insbesondere steigt der Recall-Wert für das Attribut *Voraussetzung* um 73 Prozentpunkte auf 98%, was bestätigt, dass bei der Strukturierung die Voraussetzungen häufig nicht als zu klassifizierendes Segment identifiziert werden. Die Verbesserung der weiteren Werte gegenüber der Strukturierung lässt sich mit einer teils fehlerhaften Segmentierung im Rahmen der Strukturierung erklären. So werden Textsegmente erkannt und anschließend klassifiziert, die keinem der drei Freitextattribute zugehörig sind.

Tabelle 30: Ergebnisse der Freitextklassifikation (in %)

Attribut	Recall	Precision	F1
Ziele	84	89	86
Beschreibung	79	84	81
Voraussetzungen	98	92	95

8.4 FAZIT

In diesem Kapitel wurde ein Konzept vorgestellt, um Textdokumente in eine strukturierte Form zu überführen. Das Konzept erlaubt es, unterschiedliche Methoden zur Identifikation einzelner Attribute miteinander zu kombinieren. Dafür muss zunächst eine Reihenfolge für die Identifikation der einzelnen Attribute definiert werden. In Abhängigkeit des jeweiligen Attributs wird dann nach dessen Identifikation der entsprechende Attributwert aus dem zu strukturierenden Dokument entfernt.

Das vorgestellte Konzept wurde im Rahmen einer Fallstudie prototypisch in der Domäne *Ausschreibungen studentischer Abschlussarbeiten* umgesetzt und mittels eines manuell erstellten Datensatzes evaluiert. Es zeigte sich dabei die grundsätzliche Nutzbarkeit des Konzeptes zur Strukturierung. Die prototypische Umsetzung weist allerdings Schwächen auf. Insbesondere hat sich gezeigt, dass durch die im Prototyp verwendete Segmentierung bei der Identifikation der Freitextattribute mittels fester Segmente fehlerhafte Segmente identifiziert werden. Dies kann in der anschließenden Klassifikation der Freitextattribute einen negativen Einfluss auf die Ergebnisse haben. Eine weitere Evaluation konnte jedoch zeigen, dass bei erfolgreicher Segmentierung auch Freitextattribute zuverlässig identifiziert werden können.

9

ZUSAMMENFASSUNG UND AUSBLICK

9.1 ZUSAMMENFASSUNG UND BEITRÄGE DER ARBEIT

Ziel der Arbeit war die Konzeption und Untersuchung von Verfahren zur Strukturierung von Dokumenten aus spezifischen Domänen. Dabei lag der Fokus auf der Übertragbarkeit der Verfahren von einer Domäne in weitere Domänen, wobei der manuelle Aufwand gering gehalten werden sollte, bei gleichzeitig hoher Güte der Verfahren. Aus der Heterogenität der Quellen der einzelnen Dokumente resultiert eine hohe Heterogenität hinsichtlich Format, Länge und Inhalt der betrachteten Dokumente. Diese stellt neben der Übertragbarkeit eine große Herausforderung an die Entwicklung der Methoden dar. Um dieses Ziel zu erreichen wurden die im Folgenden beschriebenen Beiträge erbracht.

In vorigen Arbeiten wurden bereits ähnliche Zielsetzungen wie in der vorliegenden Dissertation adressiert. Im Rahmen dieser Dissertation wurde daher zunächst analysiert, welchen Fokus die einzelnen Arbeiten haben, und welche Nutzbarkeit durch diese Arbeiten zur Erfüllung der in dieser Dissertation adressierten Ziele gegeben ist. Dabei hat sich gezeigt, dass einzelne relevante Teilaspekte, wie beispielsweise die Erkennung von Eigennamen, bereits ausreichend untersucht wurden, aber weitere Aspekte, wie beispielsweise die domänenadaptive Erkennung von Meta-Attributen, nicht ausreichend betrachtet wurden. Weiterhin konnte kein existierendes Verfahren identifiziert werden, welches eine domänenadaptive Strukturierung von heterogenen Dokumenten erlaubt.

Als Ausgangspunkt für die weitere Arbeit wurde untersucht, welche Abhängigkeiten zwischen den einzelnen für eine Dokumentenstrukturierung relevanten Konzepten existieren. Diese Abhängigkeiten wurden in einem domänenunabhängigen Modell repräsentiert, welches festlegt, in welchen Relationen Konzepte, wie die in den Dokumenten beschriebenen Entitäten oder die die Entitäten beschreibenden Attribute, zueinander stehen.

Es wurden fünf Anwendungsdomänen, die eine hohe Heterogenität hinsichtlich unterschiedlicher Eigenschaften wie Länge, Format oder enthaltener Attribute aufweisen, ausgewählt und mittels des zuvor definierten Modells beschrieben. Die einzelnen Anwendungsdomänen wurden für die Konzeption und Evaluation der entwickelten Verfahren herangezogen. Hierbei hat sich gezeigt, dass gewisse Attributtypen in mehreren Anwendungsdomänen existieren.

Im Rahmen der Arbeiten wurden Verfahren zur Identifikation von drei dieser Attributtypen vorgestellt:

- Freitextattribute zeichnen sich durch eine variable Länge und eine sehr heterogene Struktur aus. Zu deren Identifikation wurde ein Verfahren vorgestellt, in welchem zunächst eine Segmentierung des Textes erfolgt, um anschließend die einzelnen Segmente zu klassifizieren. Zur Klassifikation wurde ein überwachtes Lernverfahren in Verbindung mit unterschiedlichen domänen-

unabhängigen Textmerkmalen verwendet. Durch Evaluation in zwei Anwendungsdomänen konnte die Eignung dieses Verfahrens zur Klassifikation von Freitextattributen in verschiedenen Domänen gezeigt werden.

- Im Gegensatz zu den Freitextattributen sind die Werte der Meta-Attribute nicht wörtlich in den Dokumenten zu finden, sondern stellen vordefinierte Klassen dar. Es wurde ein Verfahren konzipiert, welches die Zugehörigkeit eines Dokumentes zu einer Klasse mittels des kompletten im Dokument enthaltenen Textes bestimmt. Bei diesem Klassifikationsverfahren handelt es sich um ein maschinelles Lernverfahren, was mit weniger annotierten Trainingsdaten als klassische überwachte Lernverfahren auskommt. Die Innovation dieses Verfahrens besteht darin, dass mittels der Technik des Active Learning geeignete Daten zur manuellen Annotation ausgewählt werden, so dass keine Daten ohne hohen Informationsgehalt für das Training des Klassifikators annotiert werden, und Ensemble Learning zur Erzielung einer hohen Klassifikationsgüte eingesetzt wird. In einer Evaluation konnte gezeigt werden, dass die angestrebten Eigenschaften des Verfahrens erfolgreich erreicht wurde und gleichzeitig eine bessere Performanz als bei vergleichbaren Verfahren erzielt wird. Der Aufwand für die Nutzung des Verfahrens in unterschiedlichen Domänen ist, aufgrund der geringeren notwendigen Menge an Trainingsdaten, geringer als bei bekannten Verfahren.

- Zur Identifikation aggregierter Attribute, die aus mehreren atomaren Attributen bestehen, wurde ein Konzept vorgestellt, welches die Identifikation unter Verwendung übertragbarer Regeln und unter Einbindung domänenübergreifender Wissensquellen erlaubt. Am Beispiel der Identifikation von postalischen Unternehmensadressen wurde das Verfahren evaluiert. Dabei konnten deutlich bessere Ergebnisse als bei Ansätzen verwandter Arbeiten erzielt werden.

Die entwickelten und evaluierten Verfahren zeichnen sich insgesamt dadurch aus, dass der Aufwand für die Übertragung in verschiedene Domänen geringer als bei bestehenden Ansätzen ist, zugleich aber eine vergleichbar hohe Güte der Verfahren erreicht wird.

Um das Gesamtziel dieser Dissertation, die Strukturierung kompletter Textdokumente, zu erreichen, ist es notwendig, Methoden zur Identifikation einzelner Attribute miteinander zu kombinieren. Zur Realisierung dieser Kombination wurde ein generisches Konzept vorgestellt, welches durch einfache Regeln das Zusammenspiel der einzelnen Methoden ermöglicht. Das Konzept wurde prototypisch in einer Anwendungsdomäne umgesetzt und evaluiert. Es konnte gezeigt werden, dass das Konzept eine Strukturierung von Dokumenten ermöglicht, indem die den Dokumenten zu entnehmenden Attributwerte in strukturierter Form identifiziert werden. Die resultierende strukturierte Repräsentation ist eine zentrale Voraussetzung für die Realisierung weitergehender Anwendungen wie domänenspezifische Suchmaschinen oder Empfehlungssysteme, die helfen, die bei der Informationsbeschaffung aus Dokumenten, insbesondere des Internets, auftretenden Schwierigkeiten zu reduzieren.

9.2 AUSBLICK

Die in dieser Dissertation entworfenen Methoden bieten die Grundlage für weitergehende Arbeiten. Zugleich bestehen Möglichkeiten der Verfeinerung und Weiterentwicklung der entwickelten Verfahren zur Identifikation der einzelnen Attribute. Auf einige weiter zu adressierende Aspekte wird im Folgenden eingegangen.

Die vorgestellte Identifikation der Freitextattribute basiert auf einem manuell zu definierendem Segmentierungsschritt. Diese Segmentierung könnte zusätzlich automatisiert auf Basis des textuellen Inhalts umgesetzt werden (siehe verwandte Arbeiten in Abschnitt 3.1.2), um die Abhängigkeit von festen Segmentgrenzen wie Satzzeichen oder Zeilenumbrüchen zu reduzieren. Die Ergebnisse in den verwandten Arbeiten lassen bei längeren Segmenten gute Ergebnisse erwarten. Bei kurzen Segmenten könnte ein hybrider Ansatz, welcher sowohl inhaltliche Merkmale als auch strukturelle Merkmale der Dokumente zur Segmentierung nutzt, zielführend sein. Zur Klassifikation sollte weiterhin die Klassifikationsgüte bei Verwendung sequentieller Klassifikationsmodelle wie den Conditional Random Fields (CRF) untersucht werden. Da bei der Klassifikation von Sätzen in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* die hohe Relevanz des Merkmals der relativen Satzposition zu erkennen war, ist zu erwarten, dass solch ein sequentielles Modell hier sehr gute Ergebnisse liefern könnte, welches diese Abhängigkeiten berücksichtigt [1]. Weiterhin könnte die Identifikation von Freitextattributen mit einer Analyse der Argumentationsstrukturen (siehe beispielsweise [79]) kombiniert werden, um Abhängigkeiten zwischen Sätzen zu identifizieren.

Für das vorgestellte Verfahren zur Identifikation der Meta-Attribute sollte das Verhalten des Klassifikationssystems bei längerfristiger Nutzung und regelmäßigem Neu-Training des Spezialklassifikators untersucht werden. Insbesondere wäre es hier von Interesse, zu analysieren, wann es vorteilhaft sein könnte, den Basisklassifikator mit den gesammelten Trainingsdaten erneut zu trainieren, da das System einen stabilen Zustand erreicht hat.

In der vorgestellten Arbeiten wurden die zur Identifkation von aggregierten Attributen und zur Kombination der Identifikationsverfahren verwendeten Parameter zur Definition maximal zulässiger Tokenabstände und Längen von Tokensequenzen sowie minimaler Attributwertlängen auf Basis einer manuellen Analyse von Beispieldokumenten festgelegt. Eine optimale Wahl dieser Parameter ließe sich auch domänenabhängig mittels eines maschinellen Lernverfahrens bestimmen.

Die mittels der Verfahren strukturierten Daten könnten in zahlreichen Anwendungen genutzt werden. So sollte untersucht werden, wie die Nutzung der nach Einsatz der in dieser Arbeit vorgestellten Strukturierung für die Umsetzung einer facettierten Suche in einer oder verschiedenen Anwendungsdomänen erfolgen kann. Weiterhin wäre in Benutzerstudien zu evaluieren, welchen Mehrwert eine solche Form der Suche dem Nutzer gegenüber einer Volltextsuche bringt und wie stark die Informationsüberflutung so reduziert werden kann. Ebenso gilt es zu untersuchen, wie Empfehlungssysteme auf Basis der strukturierten Daten aussehen könnten und welchen Mehrwert sie für den Nutzer haben.

Das Konzept dieser Arbeit sieht vor, dass die Attributwerte, abgesehen der Meta-Attribute, jeweils wörtliche Elemente des ursprünglichen Textes sind. Eine Kanonisierung der Attributwerte oder sogar eine semantische Analyse der Attributwerte würde weitere Anwendungen ermöglichen. Insbesondere bei einer Verknüpfung der

kanonisierten Attributwerte mit Konzepten aus Ontologien könnten unter Verwendung von Inferenzsystemen intelligente Anwendungen entwickelt werden. So könnten beispielsweise Ähnlichkeiten zwischen strukturierten Dokumenten ohne Verwendung statistischer Verfahren bestimmt werden.

LITERATURVERZEICHNIS

[1] Aggarwal, Charu C. (Hrsg.) ; Zhai, ChengXiang (Hrsg.): *Mining Text Data*. New York Dordrecht Heidelberg London : Springer, 2012. – ISBN 978-1-4614-3223-4

[2] Aggarwal, Charu C. ; Zhai, ChengXiang: A Survey of Text Classification Algorithms. In: Aggarwal, Charu C. (Hrsg.) ; Zhai, ChengXiang (Hrsg.): *Mining Text Data*. New York Dordrecht Heidelberg London : Springer, 2012, S. 163–222. – ISBN 978-1-4614-3223-4

[3] Ahlers, Dirk ; Boll, Susanne: Retrieving Address-Based Locations from the Web. In: *Proceedings of the 2nd international workshop on Geographic Information Retrieval*, 2008

[4] Álvarez, Manuel ; Pan, Alberto ; Raposo, Juan ; Bellas, Fernando ; Cacheda, Fidel: Extracting Lists of Data Records from Semi-structured Web Pages. In: *Elsevier Data & Knowledge Engineering* 64 (2008), Nr. 2, S. 491–509

[5] Anjorin, Mojisola ; Rensing, Christoph ; Bischoff, Kerstin ; Bogner, Christian ; Lehmann, Lasse ; Reger, Anna L. ; Faltin, Nils ; Steinacker, Achim ; Lüdemann, Andy ; Domínguez García, Renato: CROKODIL - A Platform for Collaborative Resource-Based Learning. In: *Towards Ubiquitous Learning*. Berlin Heidelberg : Springer, 2011 (Lecture Notes in Computer Science 6964), S. 29–42. – ISBN 978-3-642-23984-7

[6] Asadi, Saeid ; Yang, Guowei ; Zhou, Xiaofang ; Shi, Yuan ; Zhai, Boxuan ; Jiang, WendyWen-Rong: Pattern-Based Extraction of Addresses from Web Page Content. In: *Progress in WWW Research and Development* Bd. 4976. Springer Berlin Heidelberg, 2008, S. 407–418. – ISBN 978-3-540-78848-5

[7] Baccianella, Stefano ; Esuli, Andrea ; Sebastiani, Fabrizio: SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation*, 2010

[8] Banko, Michele ; Cafarella, Michael J. ; Soderland, Stephen ; Broadhead, Matt ; Etzioni, Oren: Open Information Extraction from the Web. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007

[9] Barkschat, Kai: Semantic Information Extraction on Domain Specific Data Sheets. In: *The Semantic Web: Trends and Challenges*. Berlin Heidelberg : Springer, Mai 2014 (Lecture Notes in Computer Science 8465), S. 864–873. – ISBN 978-3-319-07442-9

[10] Bateman, John ; Kamps, Thomas ; Kleinz, Jörg ; Reichenberger, Klaus: Towards Constructive Text, Diagram, and Layout Generation for Information Presentation. In: *Computational Linguistics* 27 (2001), Nr. 3, S. 409–449

[11] Bawden, David ; Robinson, Lyn: The Dark Side of Information: Overload, Anxiety and other Paradoxes and Pathologies. In: *Journal of Information Science* 35 (2009), Nr. 2, S. 180–191

[12] Beeferman, Doug ; Berger, Adam ; Lafferty, John: Text Segmentation Using Exponential Models. In: *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 1997

[13] Benikova, Darina ; Biemann, Chris ; Kisselew, Max ; Padó, Sebastian: GermEval 2014 Named Entity Recognition Shared Task: Companion Paper. In: *Proceedings of the KONVENS GermEval Shared Task on Named Entity Recognition*, 2014

[14] Berners-Lee, Tim ; Hendler, James ; Lassila, Ora: The Semantic Web. In: *Scientific American* 284 (2001), Nr. 5, S. 34–43

[15] Bishop, Christopher: *Pattern Recognition and Machine Learning*. 1. Ausgabe 2006, korrigierter 2. Druck 2011. New York : Springer, 2007. – ISBN 978-0-387-31073-2

[16] Blei, David M.: Probabilistic Topic Models. In: *Communications of the ACM* 55 (2012), April, Nr. 4, S. 77–84

[17] Blei, David M. ; Ng, Andrew Y. ; Jordan, Michael I.: Latent Dirichlet Allocation. In: *Journal of Machine Learning Research* 3 (2003), S. 993–1022

[18] Blitzer, John ; Dredze, Mark ; Pereira, Fernando: Biographies, Bollywood, Boom-Boxes and Blenders: Domain Adaptation for Sentiment Classification. In: *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Juni 2007

[19] Blumberg, Robert ; Atre, Shaku: The Problem with Unstructured Data. In: *Data Management Review* 13 (2003), Nr. 4, S. 42–49

[20] Bollacker, Kurt ; Evans, Colin ; Paritosh, Praveen ; Sturge, Tim ; Taylor, Jamie: Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA, 2008

[21] Brants, Thorsten ; Chen, Francine ; Farahat, Ayman: A System for New Event Detection. In: *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, 2003

[22] Breiman, Leo: Bagging Predictors. In: *Machine Learning* 24 (1996), Nr. 2, S. 123–140

[23] Bridgwater, Adrian: *IBM: 80 percent of our global data is unstructured (so what do we do?)*. Oktober 2010. – URL http://www.computerweekly.com/blogs/cwdn/2010/10/ibm-80-percent-of-data-is-unstructured-so-what-do-we-do.html. – Zugriffsdatum: 2015-07-09. – Blogpost

[24] Bundesministerium der Justiz und für Verbraucherschutz: *Telemediengesetz vom 26. Februar 2007 (BGBl. I S. 179), das zuletzt durch Artikel 4 des Gesetzes vom 17. Juli 2015 (BGBl. I S. 1324) geändert worden ist*. März 2007. – Gesetz

[25] Cai, Deng ; Yu, Shipeng ; Wen, Ji-Rong ; Ma, Wei-Ying: Extracting Content Structure for Web Pages Based on Visual Representation. In: *Web Technologies and Applications*. Springer Berlin Heidelberg, 2003 (Lecture Notes in Computer Science 2642), S. 406–417. – ISBN 978-3-540-02354-8

[26] Cai, Wentao ; Wang, Shengrui ; Jiang, Qingshan: Address Extraction: Extraction of Location-Based Information from the Web. In: *Web Technologies Research and Development - APWeb 2005* Bd. 3399. Springer Berlin Heidelberg, 2005, S. 925–937. – ISBN 978-3-540-25207-8

[27] Cavnar, William ; Trenkle, John: N-Gram-Based Text Categorization. In: *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval*, 1994

[28] Chang, Chia H. ; Kayed, Mohammed ; Girgis, Moheb R. ; Shaalan, Khaled F.: A Survey of Web Information Extraction Systems. In: *IEEE Transactions on Knowledge and Data Engineering* 18 (2006), Nr. 10, S. 1411–1428

[29] Choi, Freddy Y. Y.: Advances in Domain Independent Linear Text Segmentation. In: *Proceedings of the 1st North American Chapter of the Association for Computational Linguistics Conference*, 2000

[30] Cole, Charles: A Theory of Information Need for Information Retrieval That Connects Information to Knowledge. In: *Journal of the American Society for Information Science and Technology* 62 (2011), Nr. 7, S. 1216–1231

[31] Crescenzi, Valter ; Mecca, Giansalvatore ; Merialdo, Paolo: RoadRunner: Towards Automatic Data Extraction from Large Web Sites. In: *Proceedings of the 27th International Conference on Very Large Data Bases*, 2001

[32] Cristani, Matteo ; Cuel, Roberta: A Survey on Ontology Creation Methodologies. In: *International Journal on Semantic Web and Information Systems* 1 (2005), Nr. 2, S. 49–69

[33] Cucerzan, Silviu: Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In: *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007

[34] Damljanovic, Danica ; Bontcheva, Kalina: Named Entity Disambiguation using Linked Data. In: *Proceedings of the 9th Extended Semantic Web Conference*, 2012

[35] De Marneffe, Marie-Catherine ; MacCartney, Bill ; Manning, Christopher D.: Generating Typed Dependency Parses from Phrase Structure Parses. In: *Proceedings of the 5th International Conference on Language Resources and Evaluation*, 2006

[36] Déjean, Hervé ; Meunier, Jean-Luc: Structuring Documents According to Their Table of Contents. In: *Proceedings of the 2005 ACM Symposium on Document Engineering*, 2005

[37] Déjean, Hervé ; Meunier, Jean-Luc: A System for Converting PDF Documents into Structured XML Format. In: Bunke, Horst (Hrsg.) ; Spitz, A. L. (Hrsg.): *Document Analysis Systems VII*. Springer Berlin Heidelberg, 2006 (Lecture Notes in Computer Science 3872), S. 129–140. – ISBN 978-3-540-32140-8

[38] Dietterichl, Thomas G.: Ensemble Learning. In: Arbib, Michael A. (Hrsg.): *The Handbook of Brain Theory and Neural Networks*. Cambridge : MIT Press, 2002, S. 405–408. – ISBN 978-0-262-01197-6

[39] Domínguez García, Renato: *Unterstützung des Ressourcen-basierten Lernens in Online Communities - Automatische Erstellung von Grosstaxonomien in verschiedenen Sprachen*. Darmstadt, Technische Universität Darmstadt, Dissertation, 2013

[40] Domínguez García, Renato ; Schmidt, Sebastian ; Rensing, Christoph ; Steinmetz, Ralf: Automatic Taxonomy Extraction in Different Languages using Wikipedia and Minimal Language-Specific Information. In: *Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics*, 2012

[41] Dong, Yan-Shi ; Han, Ke-Song: A Comparison of Several Ensemble Methods for Text Categorization. In: *Proceedings of the 2004 IEEE International Conference on Services Computing*, 2004

[42] Du, Lan ; Buntine, Wray ; Johnson, Mark: Topic Segmentation with a Structured Topic Model. In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2013

[43] Embley, David W. ; Campbell, Douglas M. ; Smith, Randy D. ; Liddle, Stephen W.: Ontology-Based Extraction and Structuring of Information from Data-rich Unstructured Documents. In: *Proceedings of the Seventh International Conference on Information and Knowledge Management*, 1998

[44] Erbs, Nicolai: *Approaches to Automatic Text Structuring*. Darmstadt, Technische Universität Darmstadt, Dissertation, 2015

[45] Etzioni, Oren ; Cafarella, Michael ; Downey, Doug ; Kok, Stanley ; Popescu, Ana-Maria ; Shaked, Tal ; Soderland, Stephen ; Weld, Daniel S. ; Yates, Alexander: Web-Scale Information Extraction in KnowItAll (Preliminary Results). In: *Proceedings of the 13th International Conference on World Wide Web*, 2004

[46] Etzioni, Oren ; Cafarella, Michael ; Downey, Doug ; Popescu, Ana-Maria ; Shaked, Tal ; Soderland, Stephen ; Weld, Daniel S. ; Yates, Alexander: Unsupervised Named-Entity Extraction from the Web: An Experimental Study. In: *Elsevier Artificial Intelligence* 165 (2005), Nr. 1, S. 91–134

[47] Etzioni, Oren ; Fader, Anthony ; Christensen, Janara ; Soderland, Stephen ; Mausam, Mausam: Open Information Extraction: The Second Generation. In: *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011

[48] Euzenat, Jérôme ; Shvaiko, Pavel: *Ontology Matching*. Berlin, Heidelberg : Springer, 2013. – ISBN 978-3-642-38720-3

[49] Faatz, Andreas ; Steinmetz, Ralf: Ontology Enrichment with Texts from the WWW. In: *Proceedings of ECML - 2nd Workshop Semantic Web Mining*, 2002

[50] Finn, Aidan ; Kushmerick, Nicholas ; Smyth, Barry: Genre Classification and Domain Transfer for Information Filtering. In: *Advances in Information Retrieval*. Berlin Heidelberg : Springer, 2002 (Lecture Notes in Computer Science 2291), S. 353–362. – ISBN 978-3-540-43343-9

[51] Foundation, Wikimedia: *Wikipedia: List of infoboxes*. 2015. – URL https://en.wikipedia.org/w/index.php?title=Wikipedia:List_of_infoboxes&oldid=679958485. – Zugriffsdatum: 2015-09-09. – Webseite

[52] Fu, Yifan ; Zhu, Xingquan ; Li, Bin: A Survey on Instance Selection for Active Learning. In: *Springer Knowledge and Information Systems* 35 (2013), Nr. 2, S. 249–283

[53] Fürnkranz, Johannes: A Study Using n-Gram Features for Text Categorization / Austrian Research Institute for Artifical Intelligence. 1998 (3). – Forschungsbericht. Technischer Bericht

[54] Gamma, Erich ; Helm, Richard ; Johnson, Ralph ; Vlissides, John: *Design Patterns: Elements of Reusable Object-Oriented Software*. Upper Saddle River : Prentice Hall, 1995. – ISBN 978-0-201-63361-0

[55] Gangemi, Aldo: A Comparison of Knowledge Extraction Tools for the Semantic Web. In: *The Semantic Web: Semantics and Big Data*. Berlin Heidelberg : Springer, 2013 (Lecture Notes in Computer Science 7882), S. 351–366. – ISBN 978-3-642-38287-1

[56] Ganz, John F. ; Chute, Christopher ; Manfrediz, Alex ; Minton, Stephen ; Reinsel, David ; Schlichting, Wolfgang ; Toncheva, Anna: The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth Through 2011 / IDC: Analyze the Future. 2008. – Forschungsbericht

[57] Glorot, Xavier ; Bordes, Antoine ; Bengio, Yoshua: Domain Adaptation for Large-Scale Sentiment Classification: A Deep Learning Approach. In: *Proceedings of the 28th International Conference on Machine Learning*, 2011

[58] Grishman, Ralph: Information Extraction: Techniques and Challenges. In: *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*. Berlin Heidelberg : Springer, 1997 (Lecture Notes in Computer Science 1299), S. 10–27. – ISBN 978-3-540-63438-6

[59] Gröchenig, Simon ; Brunauer, Richard ; Rehrl, Karl: Estimating Completeness of VGI Datasets by Analyzing Community Activity Over Time Periods. In: *Connecting a Digital Europe Through Location and Place*. Berlin Heidelberg : Springer, 2014 (Lecture Notes in Geoinformation and Cartography), S. 3–18. – ISBN 978-3-319-03610-6

[60] Guo, Yufan ; Korhonen, Anna ; Liakata, Maria ; Karolinska, Ilona S. ; Sun, Lin ; Stenius, Ulla: Identifying the Information Structure of Scientific Abstracts: An Investigation of Three Different Schemes. In: *Proceedings of the 2010 Workshop on Biomedical Natural Language Processing*, 2010

[61] Hacimüftüoglu, Volkan: *Bereitstellung von strukturierten Daten für eine Suchmaschine für Ausschreibungen studentischer Abschlussarbeiten*. Darmstadt, Technische Universität Darmstadt, Bachelorarbeit, 2014

[62] Haklay, Mordechai ; Weber, Patrick: OpenStreetMap: User-Generated Street Maps. In: *IEEE Pervasive Computing* 7 (2008), Nr. 4, S. 12–18

[63] Hall, Mark ; Frank, Eibe ; Holmes, Geoffrey ; Pfahringer, Bernhard ; Reutemann, Peter ; Witten, Ian H.: The WEKA Data Mining Software: An Update. In: *ACM SIGKDD Explorations* 11 (2009), Nr. 1, S. 10–18

[64] Han, Xianpei ; Zhao, Jun: Named Entity Disambiguation by Leveraging Wikipedia Semantic Knowledge. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009

[65] Hearst, Marti A.: Automatic Acquisition of Hyponyms from Large Text Corpora. In: *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, 1992

[66] Hearst, Marti A.: TextTiling: Segmenting Text into Multi-Paragraph Subtopic Passages. In: *Computational Linguistics* 23 (1997), Nr. 1, S. 33–64

[67] Hearst, Marti A. ; Plaunt, Christian: Subtopic Structuring for Full-length Document Access. In: *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1993

[68] Hilbert, Martin ; López, Priscila: The World's Technological Capacity to Store, Communicate, and Compute Information. In: *Science* 332 (2011), Nr. 6025, S. 60–65

[69] Hoffart, Johannes ; Yosef, Mohamed A. ; Bordino, Ilaria ; Fürstenau, Hagen ; Pinkal, Manfred ; Spaniol, Marc ; Taneva, Bilyana ; Thater, Stefan ; Weikum, Gerhard: Robust Disambiguation of Named Entities in Text. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011

[70] Hong, Jer L.: Data Extraction for Deep Web Using WordNet. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews* 41 (2011), Nr. 6, S. 854–868

[71] Hovy, Eduard ; Lavid, Julia: Towards a "Science" of Corpus Annotation: A New Methodological Challenge for Corpus Linguistics. In: *International Journal of Translation* 22 (2010), Nr. 1, S. 13–36

[72] Jiang, Jing: Information Extraction from Text. In: Aggarwal, Charu C. (Hrsg.) ; Zhai, ChengXiang (Hrsg.): *Mining Text Data*. New York Dordrecht Heidelberg London : Springer, 2012, S. 11–41. – ISBN 978-1-4614-3222-7

[73] Joachims, Thorsten: A Statistical Learning Model of Text Classification for Support Vector Machines. In: *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001

[74] John, George H. ; Langley, Pat: Estimating Continuous Distributions in Bayesian Classifiers. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995

[75] Jurafsky, Dan ; Martin, James H.: *Speech and Language Processing*. 2. Auflage. Upper Saddle River, NJ : Prentice Hall, 2008. – ISBN 978-0-13-504196-3

[76] Kallipolitis, Leonidas ; Karpis, Vassilis ; Karali, Isambo: Semantic Search in the World News Domain using Automatically Extracted Metadata Files. In: *Knowledge-Based Systems* 27 (2012), März, S. 38–50

[77] Kecman, Vojislav: Support Vector Machines - An Introduction. In: Wang, Lipo (Hrsg.): *Support Vector Machines: Theory and Applications*. Berlin Heidelberg : Springer, 2005, S. 1–47. – ISBN 978-3-540-24388-5

[78] Keerthi, S. S. ; Shevade, S. K. ; Bhattacharyya, C. ; Murthy, K. R. K.: Improvements to Platt's SMO Algorithm for SVM Classifier Design. In: *Neural Computation* 13 (2001), Nr. 3, S. 637–649

[79] Kirschner, Christian ; Eckle-Kohler, Judith ; Gurevych, Iryna: Linking the Thoughts: Analysis of Argumentation Structures in Scientific Publications. In: *Proceedings of the 2nd Workshop on Argumentation Mining held in Conjunction with the 2015 Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies*, Juni 2015

[80] Koehn, Philipp: *Statistical Machine Translation*. 1. Auflage. Cambridge New York : Cambridge University Press, 2010. – ISBN 978-0-521-87415-1

[81] Kohavi, Ron: A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In: *Proceedings of the 1995 International Joint Conference on Artificial Intelligence*, 1995

[82] Kohlschütter, Christian ; Nejdl, Wolfgang: A Densitometric Approach to Web Page Segmentation. In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, 2008

[83] Kottmann, Jörn: *Apache OpenNLP*. 2010. – URL http://opennlp.apache.org/index.html. – Zugriffsdatum: 2015-08-11. – Webseite

[84] Kozima, Hideki: Text Segmentation Based on Similarity Between Words. In: *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, 1993

[85] Lafferty, John ; McCallum, Andrew ; Pereira, Fernando: Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In: *Proceedings of the 18th International Conference on Machine Learning*, 2001

[86] Lewis, David D. ; Gale, William A.: A Sequential Algorithm for Training Text Classifiers. In: *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994

[87] Li, Chenliang ; Weng, Jianshu ; He, Qi ; Yao, Yuxia ; Datta, Anwitaman ; Sun, Aixin ; Lee, Bu-Sung: TwiNER: Named Entity Recognition in Targeted Twitter Stream. In: *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2012

[88] Li, Mingkun ; Sethi, I.K.: Confidence-Based Active Learning. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), Nr. 8, S. 1251–1261

[89] Li, Xirong ; Snoek, Cees G.: Classifying Tag Relevance with Relevant Positive and Negative Examples. In: *Proceedings of the 21st ACM International Conference on Multimedia*, 2013

[90] Lin, Chin-Yew ; Hovy, Eduard: The Automated Acquisition of Topic Signatures for Text Summarization. In: *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*, 2000

[91] Liu, Bing ; Zhang, Lei: A Survey of Opinion Mining and Sentiment Analysis. In: Aggarwal, Charu C. (Hrsg.) ; Zhai, ChengXiang (Hrsg.): *Mining Text Data*. New York Dordrecht Heidelberg London : Springer, 2012, S. 415–463. – ISBN 978-1-4614-3222-7

[92] Liu, Xiaohua ; Zhou, Ming ; Wei, Furu ; Fu, Zhongyang ; Zhou, Xiangyang: Joint Inference of Named Entity Recognition and Normalization for Tweets. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, 2012

[93] Loos, Berenike ; Biemann, Chris: Supporting Web-Based Address Extraction with Unsupervised Tagging. In: *Data Analysis, Machine Learning and Applications*. Berlin Heidelberg : Springer, 2008 (Studies in Classification, Data Analysis, and Knowledge Organization), S. 577–584. – ISBN 978-3-540-78239-1

[94] Lovins, Julie: Development of a Stemming Algorithm. In: *Mechanical Translation and Computational Linguistics* 11 (1968), S. 22–31

[95] Luhn, Hans P.: A Statistical Approach to Mechanized Encoding and Searching of Literary Information. In: *IBM Journal of Research and Development* 1 (1957), Nr. 4, S. 309–317

[96] Malone, Robert: *Structuring Unstructured Data*. Juli 2007. – URL http://www.forbes.com/2007/04/04/teradata-solution-software-biz-logistics-cx_rm_0405data.html. – Zugriffsdatum: 2015-07-09. – Blogpost

[97] Manning, Christopher ; Raghavan, Prabhakar ; Schütze, Hinrich: *Introduction to Information Retrieval*. 1. Ausgabe. Cambridge : Cambridge University Press, 2008

[98] Manning, Christopher ; Schuetze, Hinrich: *Foundations of Statistical Natural Language Processing*. 1. Ausgabe. Cambridge : The MIT Press, 1999. – ISBN 978-0-262-13360-9

[99] Mansour, Riham ; Refaei, Nesma ; Gamon, Michael ; Abdul-Hamid, Ahmed ; Sami, Khaled: Revisiting the Old Kitchen Sink: Do we Need Sentiment Domain

Adaptation? In: *Proceedings of Recent Advances in Natural Language Processing*, 2013

[100] Marcus, Mitchell P. ; Marcinkiewicz, Mary A. ; Santorini, Beatrice: Building a Large Annotated Corpus of English: The Penn Treebank. In: *Computational Linguistics* 19 (1993), Nr. 2, S. 313–330

[101] Marini, Joe: *Document Object Model*. 1. Ausgabe. New York : McGraw-Hill, Inc., 2002. – ISBN 978-0-07-222436-8

[102] Maynard, Diana: Metrics for Evaluation of Ontology-Based Information. In: *In WWW 2006 Workshop on Evaluation of Ontologies for the Web*, 2006

[103] Mejova, Yelena ; Srinivasan, Padmini: Crossing Media Streams with Sentiment: Domain Adaptation in Blogs, Reviews and Twitter. In: *Proceedings of the Sixth International AAAI Conference on Weblogs and Social Media*, 2012

[104] Melo, Gerard de ; Weikum, Gerhard: MENTA: Inducing Multilingual Taxonomies from Wikipedia. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010

[105] Mertins, Inge ; Moore, Roger ; Gibbon, Dafydd (Hrsg.): *Handbook of Multimodal and Spoken Dialogue Systems: Resources, Terminology and Product Evaluation*. Norwell, MA, USA : Kluwer Academic Publishers, 2000. – ISBN 978-0-7923-7904-1

[106] Meusel, Robert ; Paulheim, Heiko: Linked Data for Information Extraction Challenge 2014 : Tasks and Results. In: *Proceedings of the Second International Workshop on Linked Data for Information Extraction*, 2014

[107] Meusel, Robert ; Petrovski, Petar ; Bizer, Christian: The WebDataCommons Microdata, RDFa and Microformat Dataset Series. In: *Proceedings of the 13th International Semantic Web Conference - Part I*, 2014

[108] Meyer, Christian M. ; Gurevych, Iryna: OntoWiktionary – Constructing an Ontology from the Collaborative Online Dictionary Wiktionary. In: Pazienza, Maria T. (Hrsg.) ; Stellato, Armando (Hrsg.): *Semi-Automatic Ontology Development: Processes and Resources*. Hershey, PA, USA : IGI Global, 2012, S. 131–161. – ISBN 978-1-4666-0188-8

[109] Miller, George A.: WordNet: A Lexical Database for English. In: *Communications of the ACM* 38 (1995), Nr. 11, S. 39–41

[110] Mitchell, Tom M.: *Machine Learning*. Boston : McGraw-Hill, 1997. – ISBN 978-0-07-115467-3

[111] Mohit, Behrang: Named Entity Recognition. In: Zitouni, Imed (Hrsg.): *Natural Language Processing of Semitic Languages*. Berlin Heidelberg : Springer, 2014 (Theory and Applications of Natural Language Processing), S. 221–245. – ISBN 978-3-642-45357-1

[112] Müller, Hans-Michael ; Kenny, Eimear E. ; Sternberg, Paul W.: Textpresso: An Ontology-Based Information Retrieval and Extraction System for Biological Literature. In: *PLOS Biology* 2 (2004), Nr. 11, S. e309

[113] Nadeau, David ; Sekine, Satoshi: A Survey of Named Entity Recognition and Classification. In: *Lingvisticae Investigationes* 30 (2007), Nr. 1, S. 3–26

[114] Navigli, Roberto: Word Sense Disambiguation: A Survey. In: *ACM Computing Surveys* 41 (2009), Nr. 2, S. 10:1–10:69

[115] Neis, Pascal ; Zielstra, Dennis ; Zipf, Alexander: The Street Network Evolution of Crowdsourced Maps: OpenStreetMap in Germany 2007–2011. In: *Future Internet* 4 (2011), Nr. 1, S. 1–21

[116] Nenkova, Ani ; McKeown, Kathleen: A Survey of Text Summarization Techniques. In: Aggarwal, Charu C. (Hrsg.) ; Zhai, ChengXiang (Hrsg.): *Mining Text Data*. New York Dordrecht Heidelberg London : Springer, 2012, S. 43–76. – ISBN 978-1-4614-3222-7

[117] Nesi, Paolo ; Pantaleo, Gianni ; Tenti, Marco: Ge(o)Lo(cator): Geographic Information Extraction from Unstructured Text Data and Web Documents. In: *Proceedings of 9th International Workshop on Semantic and Social Media Adaptation and Personalization*, 2014

[118] Normung e.V., DIN Deutsches I. für: *DIN 5008 – Schreib- und Gestaltungsregeln für die Textverarbeitung (Sonderdruck von DIN 5008:2011)*. 5. Auflage. Berlin : Beuth Verlag, 2011. – ISBN 978-3-410-21367-3

[119] Nothman, Joel ; Ringland, Nicky ; Radford, Will ; Murphy, Tara ; Curran, James R.: Learning Multilingual Named Entity Recognition from Wikipedia. In: *Artificial Intelligence* 194 (2013), S. 151–175

[120] Özmen, Can ; Streicher, Alexander ; Zielinski, Andrea: Using Text Segmentation Algorithms for the Automatic Generation of E-Learning Courses. In: *Proceedings of the Third Joint Conference on Lexical and Computational Semantics*, 2014

[121] Page, Lawrence ; Brin, Sergey ; Motwani, Rajeev ; Winograd, Terry: The PageRank Citation Ranking: Bringing Order to the Web / Stanford InfoLab. 1999. – Forschungsbericht

[122] Pan, Sinno J. ; Yang, Qiang: A Survey on Transfer Learning. In: *IEEE Transactions on Knowledge and Data Engineering* 22 (2010), Nr. 10, S. 1345–1359

[123] Pan, Weike ; Zhong, Erheng ; Yang, Qiang: Transfer Learning for Text Mining. In: Aggarwal, Charu C. (Hrsg.) ; Zhai, ChengXiang (Hrsg.): *Mining Text Data*. New York Dordrecht Heidelberg London : Springer, 2012, S. 223–257. – ISBN 978-1-4614-3222-7

[124] Patricia, Novi ; Caputo, Barbara: Learning to Learn, from Transfer Learning to Domain Adaptation: A Unifying Perspective. In: *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014

[125] Piskorski, Jakub ; Yangarber, Roman: Information Extraction: Past, Present and Future. In: Poibeau, Thierry (Hrsg.) ; Saggion, Horacio (Hrsg.) ; Piskorski,

Jakub (Hrsg.) ; YANGARBER, Roman (Hrsg.): *Multi-Source, Multilingual Information Extraction and Summarization*. Berlin Heidelberg : Springer, 2013, S. 23–49. – ISBN 978-3-642-28568-4

[126] PLAS, Lonneke van der ; TIEDEMANN, Jörg: Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity. In: *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006

[127] PLATT, John C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: SCHOELKOPF, Bernhard (Hrsg.) ; BURGES, Christopher J. (Hrsg.) ; SMOLA, Alexander J. (Hrsg.): *Advances in Kernel Methods - Support Vector Learning*. Cambridge : MIT Press, 1998, S. 41–65. – ISBN 978-0-262-19416-7

[128] POLIKAR, Robi: Ensemble Learning. In: ZHANG, Cha (Hrsg.) ; MA, Yunqian (Hrsg.): *Ensemble Machine Learning*. New York Dordrecht Heidelberg London : Springer, 2012, S. 1–34. – ISBN 978-1-4419-9325-0

[129] QUINLAN, Ross: *C4.5: Programs for Machine Learning*. San Mateo, CA : Morgan Kaufmann Publishers, 1993

[130] RATINOV, Lev ; ROTH, Dan: Design Challenges and Misconceptions in Named Entity Recognition. In: *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, 2009

[131] REICHENBERGER, Klaus ; RONDHUIS, Klaas J. ; KLEINZ, Jörg ; BATEMAN, John A.: Effective Presentation of Information Through Page Layout: A Linguistically-Based Approach. In: *Proceedings of ACM Workshop on Effective Abstractions in Multimedia, Layout and Interaction*, 1995

[132] REINHARDT, Wolfgang ; SCHMIDT, Benedikt ; SLOEP, Peter ; DRACHSLER, Hendrik: Knowledge Worker Roles and Actions—Results of Two Empirical Studies. In: *Knowledge and Process Management* 18 (2011), Nr. 3, S. 150–174. – none

[133] RIEDL, Martin ; BIEMANN, Chris: How Text Segmentation Algorithms Gain from Topic Models. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2012

[134] RITTER, Alan ; CLARK, Sam ; MAUSAM ; ETZIONI, Oren: Named Entity Recognition in Tweets: An Experimental Study. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011

[135] ROHRBACH, Marcus ; STARK, Michael ; SZARVAS, György ; GUREVYCH, Irina ; SCHIELE, Bernt: What Helps Where - And Why? Semantic Relatedness for Knowledge Transfer. In: *Proceedings of 2010 IEEE Conference on Computer Vision and Pattern Recognition*, 2010

[136] ROKACH, Lior ; MAIMON, Oded: *Data Mining with Decision Trees: Theory and Applications*. New Jersey : World Scientific Publishing Company, 2007. – ISBN 978-981-277-171-1

[137] Runkler, Thomas A.: *Data Mining: Methoden und Algorithmen intelligenter Datenanalyse*. 2010. Wiesbaden : Vieweg+Teubner Verlag, 2011. – ISBN 978-3-8348-0858-5

[138] Russell, Stuart ; Norvig, Peter: *Artificial Intelligence: A Modern Approach*. 3rd. Upper Saddle River : Prentice Hall Press, 2009. – ISBN 978-0-13-604259-4

[139] Sánchez, David ; Isern, David ; Millan, Miquel: Content Annotation for the Semantic Web: An Automatic Web-Based Approach. In: *Springer Knowledge and Information Systems* 27 (2011), Nr. 3, S. 393–418

[140] Sanoja, Andrés ; Gancarski, Stéphane: Block-o-Matic: A Web Page Segmentation Framework. In: *Proceedings of 2014 International Conference on Multimedia Computing and Systems*, 2014

[141] Sanoja, Andrés ; Gançarski, Stéphane: Web Page Segmentation Evaluation. In: *Proceedings of the 30th Annual ACM Symposium on Applied Computing*, 2015

[142] Santorini, Beatrice: Part-of-Speech Tagging Guidelines for the Penn Treebank Project (3rd Revision) / University of Pennsylvania. 1990. – MS-CIS-90-47. Technischer Bericht

[143] Sarawagi, Sunita: Information Extraction. In: *Foundations and Trends in Databases* 1 (2008), Nr. 3, S. 261–377

[144] Sbattella, Licia ; Tedesco, Roberto: A Novel Semantic Information Retrieval System Based on a Three-level Domain Model. In: *Elsevier Journal of Systems and Software* 86 (2013), Nr. 5, S. 1426–1452

[145] Schiller, Anne ; Teufel, Simone ; Thielen, Christine: Guidelines für das Tagging deutscher Textcorpora mit STTS / Universität Stuttgart und Universität Tübingen. 1995 (66). – Forschungsbericht

[146] Schmid, Helmut: Improvements In Part-of-Speech Tagging with an Application to German. In: *In Proceedings of the ACL Special Interest Group on Linguistic Data and Corpus-based Approaches to NLP Workshop*, 1995

[147] Schmidt, Sebastian ; Manschitz, Simon ; Rensing, Christoph ; Steinmetz, Ralf: Extraction of Address Data from Unstructured Text Using Free Knowledge Resources. In: *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies*, 2013

[148] Schmidt, Sebastian ; Schnitzer, Steffen ; Rensing, Christoph: Domain-independent Sentence Type Classification: Examining the Scenarios of Scientific Abstracts and Scrum Protocols. In: *Proceedings of the 14th International Conference on Knowledge Technologies and Data-driven Business*, 2014

[149] Schnitzer, Steffen: *Effective Classification of Ambiguous Web Documents Incorporating Human Feedback Efficiently*. Darmstadt, Hochschule Darmstadt, Masterarbeit, 2013

[150] Schnitzer, Steffen ; Schmidt, Sebastian ; Rensing, Christoph ; Harriehausen-Mühlbauer, Bettina: Combining Active and Ensemble Learning for Efficient Classification of Web Documents. In: *Polibits: Research Journal on Computer Science and Computer Engineering with Applications* (2014), Nr. 49, S. 39–46. – Die beiden Erstautoren haben zu gleichen Teilen Beiträge zur Publikation geleistet.

[151] Schölkopf, Bernhard ; Smola, Alexander J.: *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge : MIT Press, 2001. – ISBN 978-0-262-19475-4

[152] Scholl, Philipp: *Semantic and Structural Analysis of Web–Based Learning Resources: Supporting Self–Directed Resource–Based Learning*. Darmstadt, Technische Universität Darmstadt, Dissertation, 2011

[153] Sebastiani, Fabrizio: Machine Learning in Automated Text Categorization. In: *ACM Computing Surveys* 34 (2002), Nr. 1, S. 1–47

[154] Settles, Burr ; Craven, Mark ; Friedland, Lewis: Active Learning with Real Annotation Costs. In: *Proceedings of the NIPS Workshop on Cost-Sensitive Learning*, 2008

[155] Shaalan, Khaled: A Survey of Arabic Named Entity Recognition and Classification. In: *Computational Linguistics* 40 (2013), Nr. 2, S. 469–510

[156] Small, Sharon G. ; Medsker, Larry: Review of Information Extraction Technologies and Applications. In: *Neural Computing and Applications* 25 (2013), Nr. 3-4, S. 533–548

[157] Sokolova, Marina ; Lapalme, Guy: A Systematic Analysis of Performance Measures for Classification Tasks. In: *Elsevier Information Processing & Management* 45 (2009), Nr. 4, S. 427–437

[158] Song, Yangqiu ; Roth, Dan: On Dataless Hierarchical Text Classification. In: *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014

[159] Sparck Jones, Karen: A Statistical Interpretation of Term Specificity and Its Application in Retrieval. In: *Journal of Documentation* 28 (1972), Nr. 1, S. 11–21

[160] Sriram, Bharath ; Fuhry, Dave ; Demir, Engin ; Ferhatosmanoglu, Hakan ; Demirbas, Murat: Short Text Classification in Twitter to Improve Information Filtering. In: *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2010

[161] Steinmetz, Ralf: *Multimedia-Technologie: Grundlagen, Komponenten und Systeme*. 3. Auflage. Berlin : Springer, 2000. – ISBN 978-3-540-67332-3

[162] Stille, Wolfgang ; Erbs, Nicolai ; Zesch, Torsten ; Gurevych, Iryna ; Weihe, Karsten: Aufbereitung und Strukturierung von Information mittels automatischer Sprachverarbeitung. In: *Tagungsband KnowTech: Unternehmenswissen als Erfolgsfaktor mobilisieren!*, 2011

[163] Suhara, Yoshihiko ; Toda, Hiroyuki ; Nishioka, Shuichi ; Susaki, Seiji: Automatically Generated Spam Detection Based on Sentence-Level Topic Information. In: *Proceedings of the 22nd International Conference on World Wide Web*, 2013

[164] Sundblad, Håkan: Automatic Acquisition of Hyponyms and Meronyms from Question Corpora. In: *Proceedings of the Workshop on Natural Language Processing and Machine Learning for Ontology Engineering at ECAI'2002*, 2002

[165] Sutton, Charles ; McCallum, Andrew: An Introduction to Conditional Random Fields. In: *Foundations and Trends in Machine Learning* 4 (2011), Nr. 4, S. 267–373

[166] Tatar, Serhan ; Cicekli, Ilyas: Automatic Rule Learning Exploiting Morphological Features for Named Entity Recognition in Turkish. In: *Journal of Information Science* 37 (2011), Nr. 2, S. 137–151

[167] Tunkelang, Daniel: *Faceted Search*. San Francisco : Morgan and Claypool Publishers, 2009. – ISBN 978-1-59829-999-1

[168] Unicode Staff, Corporate: *The Unicode Standard: Worldwide Character Encoding*. 1. Auflage. Boston : Addison-Wesley Longman Publishing Company, 1991. – ISBN 978-0-201-56788-5

[169] VersionOne: 9th Annual State of Agile Survey / VersionOne. URL http://info.versionone.com/state-of-agile-development-survey-ninth.html, 2014. – Umfrage

[170] Wang, Tong ; Hirst, Graeme: Exploring Patterns in Dictionary Definitions for Synonym Extraction. In: *Natural Language Engineering* 18 (2012), Nr. 03, S. 313–342

[171] Wimalasuriya, Daya C. ; Dou, Dejing: Using Multiple Ontologies in Information Extraction. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, 2009

[172] Wimalasuriya, Daya C. ; Dou, Dejing: Components for Information Extraction: Ontology-Based Information Extractors and Generic Platforms. In: *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, 2010

[173] Wimalasuriya, Daya C. ; Dou, Dejing: Ontology-Based Information Extraction: An Introduction and a Survey of Current Approaches. In: *Journal of Information Science* 36 (2010), Juni, Nr. 3, S. 306–323

[174] Witten, Ian H. ; Frank, Eibe: *Data Mining: Practical Machine Learning Tools and Techniques*. 1. Ausgabe. Burlington : Morgan Kaufmann, 2011. – ISBN 978-0-12-374856-0

[175] Wöber, Karl W.: Domain-specific Search Engines. In: Fesenmaier, Daniel R. (Hrsg.) ; Wöber, Karl W. (Hrsg.) ; Werthner, Hannes (Hrsg.): *Destination Recommendation Systems: Behavioral Foundations and Applications*. Oxfordshire : CABI, 2006, S. 205–226. – ISBN 978-1-84593-109-4

[176] Wollny, Sebastian: *Methoden zur einheitlichen Strukturierung von Ausschreibungen studentischer Abschlussarbeiten*. Darmstadt, Technische Universität Darmstadt, Bachelorarbeit, September 2013

[177] Wu, Fei ; Weld, Daniel S.: Open Information Extraction Using Wikipedia. In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, 2010

[178] Yao, Xuchen ; Van Durme, Benjamin: Information Extraction over Structured Data: Question Answering with Freebase. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014

[179] Zheng, Xiaoqing ; Gu, Yiling ; Li, Yinsheng: Data Extraction from Web Pages Based on Structural-Semantic Entropy. In: *Proceedings of the 21st International Conference on World Wide Web*, 2012

ABKÜRZUNGSVERZEICHNIS

AF	Aufteilungsfaktor
AKG	Annotations-Konfidenz-Grenzwert
CENFA	Combined Ensemble and Fast Active Learner
CRF	Conditional Random Fields
DOM	Document Object Model
HMM	Hidden Markov Models
HTML	Hypertext Markup Language
NB	Naive Bayes Klassifikator
OSM	OpenStreetMap
PDF	Portable Document Format
POS	Part-of-Speech
PS	Postscript
RSSVM	Random Single Support Vector Machine
SMO	Sequential Minimal Optimization
SVM	Support Vector Machine
TF-IDF	Term Frequency - Inverse Document Frequency
Weka	Waikato Environment for Knowledge Analysis

STICHWORTVERZEICHNIS

A

ANHANG

A.1 PART-OF-SPEECH TAGS

Dieser Abschnitt gibt einen Überblick über die im Rahmen dieser Dissertation verwendeten Mengen von POS-Tags. Tabelle 31 zeigt die Tags, die für englischsprachige Texte relevant sind, während Tabelle 32 die für deutschsprachige Texte relevanten POS-Tags zeigt.

Tabelle 31: Übersicht über die englischen POS-Tags im Penn Treebank Project [100], Tabelle wurde unverändert aus der genannten Publikation übernommen, nicht dargestellt sind die POS-Tags für einzelne Sonderzeichen, Details sind [142] zu entnehmen

Tag	Beschreibung
CC	coordinating conjunction
CD	cardinal number
DT	determiner
EX	existential *there*
FW	foreign word
IN	preposition or subordinating conjunction
JJ	adjective
JJR	adjective, comparative
JJS	adjective, superlative
LS	list item marker
MD	modal
NN	noun, singular or mass
NNS	noun, plural
NNP	proper noun, singular
NNPS	proper noun, plural
PDT	predeterminer
POS	possessive ending
PRP	personal pronoun
PRP$	possessive pronoun
RB	adverb
RBR	adverb, comparative
RBS	adverb, superlative
RP	particle

Tag (Fortsetzung)	Beschreibung
SYM	symbol
TO	*to*
UH	interjection
VB	verb, base form
VBD	verb, past tense
VBG	verb, gerund or present participle
VBN	verb, past participle
VBP	verb, non-3rd person singular present
VBZ	verb, 3rd person singular present
WDT	wh-determiner
WP	wh-pronoun
WP$	possessive wh-pronoun
WRB	wh-adverb

Tabelle 32: Übersicht über die POS-Tags im Stuttgart-Tübingen-Tagset (STTS) [145], Tabelle wurde unverändert aus der genannten Publikation übernommen

Tag	Beschreibung
ADJA	attributives Adjektiv
ADJD	adverbiales oder prädikatives Adjektiv
ADV	Adverb
APPR	Präposition; Zirkumposition links
APPRART	Präposition mit Artikel
APPO	Postposition
APZR	Zirkumposition rechts
ART	bestimmter oder unbestimmter Artikel
CARD	Kardinalzahl
FM	fremdsprachliches Material
ITJ	Interjektion
KOUI	unterordnende Konjunktion mit „zu“ und Infinitiv
KOUS	unterordnende Konjunktion mit Satz
KON	nebenordnende Konjunktion
KOKOM	Vergleichspartikel, ohne Satz
NN	normales Nomen
NE	Eigennamen
PDS	substituierendes Demonstrativpronomen
PDAT	attribuierendes Demonstrativpronomen
PIS	substituierendes Indefinitpronomen

TAG (Fortsetzung)	BESCHREIBUNG
PIAT	attribuierendes Indefinitpronomen ohne Determiner
PIDAT	attribuierendes Indefinitpronomen mit Determiner
PPER	irreflexives Personalpronomen
PPOSS	substituierendes Possesivpronomen
PPOSAT	attribuierendes Possesivpronomen
PRELS	Relativpronomen substituierend
PRELAT	Relativpronomen attribuierend
PRF	reflexives Personalpronomen
PWS	substituierendes Interrogativpronomen
PWAT	attribuierendes Interrogativpronomen
PWAV	adverbiales Interrogativ- oder Relativpronomen
PAV	Pronominaladverb
PTKZU	„zu“ vor Infinitiv
PTKNEG	Negationspartikel
PTKVZ	abgetrennter Verbzusatz
PTKANT	Antwortpartikel
PTKA	Partikel bei Adjektiv oder Adverb
TRUNC	Kompositions-Erstglied
VVFIN	finites Verb, voll
VVIMP	Imperativ, voll
VVINF	Infinitiv, voll
VVIZU	Infinitiv mit „zu“, voll
VVPP	Partizip Perfekt, voll
VAFIN	finites Verb, aux
VAIMP	Imperativ, aux
VAINF	Infinitiv, aux
VAPP	Partizip Perfekt, aux
VMFIN	finites Verb, modal
VMINF	Infinitiv, modal
VMPP	Partizip Perfekt, modal
XY	Nichtwort, Sonderzeichen enthaltend
$,	Komma
$.	satzbeendende Interpunktion
$(	sonstige Satzzeichen, satzintern

A.2 INFORMATIONSGEWINN BEI DER IDENTIFIKATION VON FREITEXTATTRIBUTEN

Der *Informationsgewinn* (engl. *information gain*) eines Merkmals in Bezug auf eine Menge von Evaluationsdaten beschreibt die Effektivität, die dieses Merkmal besitzt, um zur Klassifikation einer Instanz aus dem Datensatz verwendet zu werden. Dies ist analog zur Reduktion der Entropie des Datensatzes durch das Kennen des Merkmalwertes [110]. Eine Analyse der Werte des Informationsgewinns einzelner Merkmale erlaubt somit Aussagen über die Wirksamkeit einzelner Merkmale. Idealerweise werden nur Merkmale mit einem hohen Informationsgewinn verwendet, da Merkmale mit einem niedrigen Informationsgewinn die Klassifikationsgüte höchstens geringfügig verbessern können.

Tabellen 33, 34 und 35 zeigen die zehn Merkmale mit dem höchsten Informationsgewinn für die einzelnen Evaluationskorpora. Auch wenn im Hauptteil der Dissertation (Abschnitt 5.2.3) gezeigt werden konnte, dass die Menge aller Unigramme, die am höchsten zur Klassifikationsgüte beitragenden Merkmale sind, ist in den dargestellten Tabellen zu sehen, dass in Bezug auf einzelne Merkmale der Informationsgewinn anderer Merkmale höher ist. So sticht in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* (Tabellen 33 und 34) das Merkmal der Satzposition mit Abstand auf Basis des Informationsgewinns hervor.

Insgesamt ist zu beobachten, dass die Informationsgewinne der einzelnen Merkmale in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* höher ausfallen als in der Domäne *Scrum-Protokolle*. Dies könnten einen Einfluss auf die beobachtete höhere Klassifikationsgüte in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* haben (vergleiche Abschnitt 5.2.3).

Tabelle 33: Informationsgewinn der zehn höchstgewichteten Merkmale in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* (Korpus *MM*)

Rang	Merkmal	Informationsgewinn
1	*Position*	0,629
2	*Unigramm „article“*	0,323
3	*Unigramm „this“*	0,156
4	*Unigramm „we“*	0,154
5	*Unigramm „In“*	0,131
6	*Unigramm „proposed“*	0,127
7	*Unigramm „results“*	0,121
8	*Personalpronomen*	0,103
9	*Unigramm „demonstrate“*	0,101
10	*Unigramm „show“*	0,101

Tabelle 34: Informationsgewinn der zehn höchstgewichteten Merkmale in der Domäne *Kurzfassungen wissenschaftlicher Publikationen* (Korpus *BioMed*)

Rang	Merkmal	Informationsgewinn
1	*Position*	0,717
2	*Zeitindikator (Hilfsverb Vergangenheit)*	0,162
3	*Zeit (Vergangenheit)*	0,161
4	*Unigramm „were“*	0,115
5	*Unigramm „that“*	0,107
6	*Unigramm „is“*	0,086
7	*Tokenanzahl (numerisch)*	0,079
8	*Unigramm „was“*	0,071
9	*Zeit (Gegenwart)*	0,066
10	*Zeitindikator (Endung Vergangenheit)*	0,063

Tabelle 35: Informationsgewinn der zehn höchstgewichteten Merkmale in der Domäne *Scrum-Protokolle*

Rang	Merkmal	Informationsgewinn
1	*Imperativindikator*	0,133
2	*Sentiment (gesamt)*	0,108
3	*Unigramm „should“*	0,093
4	*Sentiment (negativ)*	0,078
5	*Unigramm „need“*	0,068
6	*Zeit (Vergangenheit)*	0,058
7	*Tokenanzahl*	0,053
8	*Unigramm „can“*	0,045
9	*Unigramm „better“*	0,041
10	*Unigramm „be“*	0,041

A.3 WEITERE EVALUATIONSERGEBNISSE ZUR IDENTIFIKATION VON META-ATTRIBUTEN

Im Hauptteil der Arbeit wurden die Werte der Accuracy für das Verfahren zur Identfikation von Meta-Attributen nur für die Korpusgrößen 10% und 100% dargestellt (siehe Abschnitt 6.3.5.3). Tabelle 36 zeigt die Accuracy für alle betrachteten Korpusgrößen und alle Klassen. Weiterhin wurden für die benötigten Rechenzeiten des CENFA-Verfahrens im Hauptteil nur die Ergebnisse für 10%, 50% und 100% Korpusgröße gegeben (siehe Abschnitt 6.3.5.3). Tabelle 37 zeigt die Rechenzeiten für die weiteren evaluierten Korpusgrößen 25% und 75%.

Tabelle 36: Accuracy (in %) für alle evaluierten Korpusgrößen

Korpus-grösse	Verfahren	Klasse				
		PQS	*SE*	*TED*	*TM*	*V*
10%	CENFA	79,94	88,27	86,61	81,19	86,85
	Random	77,50	87,62	85,83	78,81	83,81
	Extended	79,94	88,57	87,92	81,19	86,37
	RSSVM	80,12	84,64	87,20	79,35	81,61
25%	CENFA	86,90	91,80	91,40	79,38	91,59
	Random	85,26	90,83	90,52	79,22	90,40
	Extended	87,56	92,18	92,32	83,96	92,13
	RSSVM	87,32	90,71	80,73	78,79	90,05
50%	CENFA	90,56	94,70	95,26	89,60	94,37
	Random	89,30	93,19	93,97	88,45	93,84
	Extended	91,53	94,55	95,84	91,60	95,12
	RSSVM	91,48	92,93	95,39	89,28	86,08
75%	CENFA	93,64	95,28	97,15	92,94	96,82
	Random	92,31	94,31	96,22	91,41	96,41
	Extended	94,68	94,76	96,97	93,89	97,05
	RSSVM	91,72	96,34	97,18	91,71	96,68
100%	CENFA	95,97	97,21	97,86	94,25	97,78
	Random	94,77	96,52	97,00	93,27	97,25
	Extended	96,63	97,11	97,98	94,87	97,98
	RSSVM	96,41	95,16	94,97	95,23	97,80

Tabelle 37: Übersicht über die benötigten Rechenzeiten in Sekunden für das Training der einzelnen Klassifikatoren, die unterschiedlichen Korpusgrößen und die Phase des Betriebs (initial = das erste Training des Systems, Neutrain. = jedes weitere Training)

	25%		75%	
	INITIAL	NEUTRAIN.	INITIAL	NEUTRAIN.
CENFA	136,0	<0,1	2909,7	<0,1
Random	136,0	<0,1	2909,7	<0,1
RSSVM	16,5	16,5	200,0	200,0
Extended	136,0	139,3	2909,7	1613,0

A.4 HEURISTIKEN ZUR IDENTIFIKATION POSTALISCHER ADRESSEN

In diesem Abschnitt werden Details zu den verwendeten Heuristiken zur Identifikation postalischer Unternehmensadressen dargestellt (Abschnitt 7.2). Eine Übersicht über die Suffixe, die zur Erkennung von Straßennamenkandidaten verwendet werden (Abschnitt 7.2.2.4), ist in Tabelle 38 zu finden. Ebenso zur Erkennung von Straßennamen werden die in Tabelle 39 dargestellten POS-Tag-Sequenzen verwendet. Zur Erkennung von Unternehmensnamenkandidaten (Abschnitt 7.2.2.5) werden die in Tabelle 40 gezeigten Indikatoren für deutsche Rechtsformen für Unternehmen verwendet.

Tabelle 38: Liste der häufigsten Suffixe deutscher Straßennamen zusammen mit ihrer Häufigkeit (mehrfach vorkommende Straßennamen wurden nur einfach gewertet)

SUFFIX	HÄUFIGKEIT	SUFFIX	HÄUFIGKEIT
straße	101.718	tal	1.388
weg	53.802	acker	1.351
berg	8.319	mühle	1.282
gasse	6.215	grund	1.264
platz	5.289	busch	1.193
hof	4.849	pfad	1.030
kamp	3.932	winkel	931
ring	3.881	heide	916
feld	3.786	stieg	911
bach	3.043	dorf	879
garten	2.250	damm	856
allee	1.935	steig	838
graben	1.784	*Summe*	213.642

Tabelle 39: Liste der POS-Tag-Sequenzen für deutsche Straßennamen unter Verwendung des Stuttgart-Tübingen-Tagsets

POS-Tag-Sequenz	Beispiel
APPR → ART → ADJA → {NN | NE | ADJA}	„An der alten Eiche“
{NN | APPR} → ART → {NN | NE | ADJA}	„An der Aalkate“
APPRART → {NN | NE} → {NN | NE}	„Am Grauen Stein“
{NN | NE} → {NN | NE}	„Lochhamer Schlag“
{APPRART | ADJA} → {NN | NE | ADJA}	„Am Kneisch“
{NN | NE | ADJA | ADV}	„Rüskenbrink“

Tabelle 40: Liste der Indikatoren für deutsche Rechtsformen

Indikator	Indikator (Fortsetzung)	Indikator (Fortsetzung)
gmbh	ev.	stiftung
ag	e.v.	eigenbetrieb
a.g.	e.v	sparkasse
a.g	vvag	ewiv
aktiengesellschaft	kgaa	se
e.k.	gag	sce
gbr	invag	egmuh
ohg	eg	bank
partgg	e.g.	
kg	reitg	

B

WISSENSCHAFTLICHE ARBEITEN DES AUTORS

B.1 ZEITSCHRIFTEN-BEITRÄGE

1. Schmidt, Sebastian ; Schnitzer, Steffen ; Rensing, Christoph: Text Classification based Filters for a Domain-Specific Search Engine. In: *Elsevier Computers in Industry* (2015). – Artikel wurde zur Publikation angenommen

2. Schnitzer, Steffen ; Schmidt, Sebastian ; Rensing, Christoph ; Harriehausen-Mühlbauer, Bettina: Combining Active and Ensemble Learning for Efficient Classification of Web Documents. In: *Polibits: Research Journal on Computer Science and Computer Engineering with Applications* (2014), Nr. 49, S. 39–45. – Die beiden Erstautoren haben zu gleichen Teilen Beiträge zur Publikation geleistet.

B.2 KONFERENZ- UND WORKSHOPBEITRÄGE

3. Schmidt, Sebastian ; Schnitzer, Steffen ; Rensing, Christoph: Generic Sentence Type Classification: Examining the Scenarios of Scientific Abstracts and Scrum Protocols. In: *Proceedings of 14th International Conference on Knowledge Technologies and Data-driven Business (i-KNOW '14)*, 2014

4. Schmidt, Sebastian ; Manschitz, Simon ; Rensing, Christoph ; Steinmetz, Ralf: Extraction of Address Data from Unstructured Text using Free Knowledge Resources. In: *Proceedings of the 13th International Conference on Knowledge Management and Knowledge Technologies, (i-KNOW '13)*, 2013. – Best Paper Award i-KNOW 2013

5. Schmidt, Sebastian ; Scholl, Philipp ; Rensing, Christoph ; Steinmetz, Ralf: Cross-Lingual Recommendations in a Resource-Based Learning Scenario. In: *Proceedings of the 6th European Conference on Technology Enhanced Learning (EC-TEL 2011)*, 2011

6. Schnitzer, Steffen ; Neitzel, Svenja ; Schmidt, Sebastian ; Rensing, Christoph: Perceived Task Similarities for Task Recommendation in Crowdsourcing Systems (zur Publikation angenommen). In: *7th International Workshop on Modeling Social Media - Behavioral Analytics in Social Media, Big Data and the Web (MSM '16)*, 2016

7. Glanz, Leonid ; Schmidt, Sebastian ; Wollny, Sebastian ; Hermann, Ben: A Vulnerability's Lifetime: Enhancing Version Information in CVE Databases. In: *Proceedings of the 15th International Conference on Knowledge Technologies and Data-driven Business (i-KNOW '15)*, 2015

8. Schnitzer, Steffen ; Rensing, Christoph ; Schmidt, Sebastian ; Borchert, Kathrin ; Hirth, Matthias ; Tran-Gia, Phuoc: Demands on Task Recommendation in Crowdsourcing Platforms - The Worker's Perspective. In: *ACM RecSys 2015 Workshop on Crowdsourcing and Human Computation for Recommender Systems (CrowdRec '15)*, 2015

9. Domínguez García, Renato ; Schmidt, Sebastian ; Rensing, Christoph ; Steinmetz, Ralf: Automatic Taxonomy Extraction in Different Languages using Wikipedia and minimal language-specific Information. In: *Proceedings of the 13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLING '13)*, 2012

C

CURRICULUM VITÆ

PERSÖNLICHE INFORMATIONEN

Name	Sebastian Schmidt
Geburtsdatum	30. August 1985
Geburtsort	Gießen
Nationalität	Deutsch

SCHUL- UND HOCHSCHULAUSBILDUNG

03/2011–12/2015	Technische Universität Darmstadt, Deutschland Fachbereich Elektrotechnik und Informationstechnik Promotionsstudium
08/2008–12/2010	Technische Universität Darmstadt, Deutschland Studium der Informatik mit Abschluss Master of Science (Anwendungsfach Bionik)
08/2009–02/2010	École polytechnique fédérale de Lausanne, Schweiz Studium der Informatik, Auslandssemester
09/2005–08/2008	Technische Universität Darmstadt, Deutschland Studium der Informatik mit Abschluss Bachelor of Science
09/2002–06/2005	Ricarda-Huch-Schule, Gießen, Deutschland Abitur
08/1992–06/2002	Brüder-Grimm-Schule, Gießen, Deutschland Grundschule und Mittelstufe

BERUFSERFAHRUNG

03/2011–heute	Technische Universität Darmstadt, Deutschland Fachgebiet *Multimedia Kommunikation* Arbeitsgruppe *Knowledge and Educational Technologies* Wissenschaftlicher Mitarbeiter
08/2008–08/2009	Technische Universität Darmstadt, Deutschland Fachgebiet *Multimedia Kommunikation* Arbeitsgruppe *Peer-to-Peer Systems* Wissenschaftliche Hilfskraft
11/2007–03/2008	Technische Universität Darmstadt, Deutschland Fachbereich Informatik Wissenschaftliche Hilfskraft, Mentor Bachelorpraktikum

AKTIVITÄTEN IN DER LEHRE

2011–2015	Betreuung von 6 Gruppen im Praktikum/Projektseminar *Multimedia Communications Lab and Project I/II*
2011–2015	Betreuung von 9 Seminararbeiten im Seminar *Current Topics in Web Applications, Information Management, and Semantics*
2011–2015	Betreuung von 9 Bachelor-, Master- und Diplomarbeiten

WISSENSCHAFTLICHES ENGAGEMENT

Management	Informationsdirektor *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* u.a. Planung der einzelnen Ausgaben sowie Jahresplanung, Verantwortlichkeit für Webseite, Ansprechpartner für alle Beteiligten, Überprüfung der formalen Rahmenbedingungen (05/2011–heute)
Gutachter	*Elsevier Pervasive and Mobile Computing*
Gutachter	*International Journal of Communication Networks and Distributed Systems*

PREISE UND AUSZEICHNUNGEN

2013–2014	Auswahl als Teilnehmer, Teilnahme und erfolgreiche Absolvierung des *Software Campus*, gefördert vom Bundesministerium für Bildung und Forschung
2013	Best Paper Award für Sebastian Schmidt, Simon Manschitz, Christoph Rensing, Ralf Steinmetz: *Extraction of Address Data from Unstructured Text using Free Knowledge Resources*. 13th International Conference on Knowledge Management and Knowledge Technologies, 2013
2011	Preis der KOM-Fördergesellschaft für die beste Masterarbeit des Fachgebiets Multimedia Kommunikation der Technischen Unversität Darmstadt

D

BETREUTE STUDENTISCHE ABSCHLUSSARBEITEN

D.1 MASTER- UND DIPLOMARBEITEN

1. Svenja Neitzel. *Semantic Relatedness of Tasks in Crowdsourcing Systems* (vorläufiger Titel). Masterarbeit (Betreuung zusammen mit Steffen Schnitzer), Technische Universität Darmstadt, geplante Abgabe März 2016.

2. Nicolas Eicke. *Efficient Text Classification by Semantic Enrichment of the Feature Space*. Masterarbeit, Technische Universität Darmstadt, Juli 2015.

3. Jiejun Wen. *Design, Implementation and Evaluation of a Dependency and Similarity Discovery Algorithm for Learning Resources based on unstructured Descriptions*. Masterarbeit (Betreuung zusammen mit Johannes Konert), Technische Universität Darmstadt, Mai 2015.

4. Steffen Schnitzer. *Effective Classification of Ambiguous Web Documents Incorporating Human Feedback Efficiently*. Masterarbeit, Hochschule Darmstadt, Oktober 2013.

5. Tamara Knierim. *Modellierung und Identifikation von relevanten textuellen Eigenschaften in Webdokumenten*. Diplomarbeit, Technische Universität Darmstadt, April 2013.

D.2 BACHELORARBEITEN

5. Tobias Michels. *Audio-based System for Detecting User Activity and Co-Participating Peers*. (Betreuung zusammen mit Irina Diaconita), Technische Universität Darmstadt, November 2014.

6. Volkan Hacimüftüoğlu. *Bereitstellung von strukturierten Daten für eine Suchmaschine für Ausschreibungen studentischer Abschlussarbeiten*. Technische Universität Darmstadt, August 2014.

7. Sebastian Wollny. *Methoden zur einheitlichen Strukturierung von Ausschreibungen studentischer Arbeiten*. Technische Universität Darmstadt, September 2013.

8. Constantin Franz. *Erweiterung eines Ansatzes zur Sentiment-Erkennung in deutschsprachigen Texten*. Technische Universität Darmstadt, August 2012.

9. Marco Ewerton. *Job Characteristics on Crowdsourcing Plattforms*. Technische Universität Darmstadt, Februar 2012.

E

ERKLÄRUNG LAUT §9 DER PROMOTIONSORDNUNG

Ich versichere hiermit, dass ich die vorliegende Dissertation allein und nur unter Verwendung der angegebenen Literatur verfasst habe.

Die Arbeit hat bisher noch nicht zu Prüfungszwecken gedient.

Darmstadt, 03. November 2015

Sebastian Schmidt

KOLOPHON

Der Schriftsatz dieser Arbeit basiert auf der von André Miede entwickelten Vorlage `classicthesis`. Diese Vorlage ist inspiriert von Robert Bringhurts Buch *"The Elements of Typographic Style"* und ist sowohl für LaTeX als auch für LyX verfügbar unter

https://www.ctan.org/pkg/classicthesis